



TUGAS AKHIR – TF 145565

**PERANCANGAN SISTEM PENGENDALIAN LAJU
ALIRAN PADA ALAT UJI KEBOCORAN PIPA
SEBAGAI PENUNJANG PRAKTIKUM SISTEM
INSTRUMENTASI INDUSTRI**

SHASKI ICRAINI DIOTAMA
NRP 10 51 15 00000 053

Dosen Pembimbing

Dr. Bambang Lelono Widjiantoro, S.T, M.T.
NIP. 196990507 199512 1 001
Ahmad Fauzan Adziimaa, S.T, M.Sc.
NPP. 1991201711052

PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNOLOGI INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018



TUGAS AKHIR – TF 145565

**PERANCANGAN SISTEM PENGENDALIAN LAJU
ALIRAN PADA ALAT UJI KEBOCORAN PIPA
SEBAGAI PENUNJANG PRAKTIKUM SISTEM
INSTURMENTASI**

SHASKI ICRAINI DIOTAMA
NRP 1051150000053

Dosen Pembimbing

Dr. Bambang Lelono Widjiantoro, S.T, M.T.
NIP. 196990507 199512 1 001
Ahmad Fauzan Adziimaa, S.T, M.Sc.
NPP. 1991201711052

PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018



FINAL PROJECT – TF 145565

***DESIGN OF FLOW CONTROL SYSTEM IN PIPING
LEAK TEST AS A PRACTICAL SUPPORT OF
INSTRUMENTATION INDUSTRY SYSTEM***

SHASKI ICRAINI DIOTAMA
NRP 10511500000053

Supervisor

Dr. Bambang Lelono Widjiantoro, S.T, M.T.
NIP. 196990507 199512 1 001
Ahmad Fauzan Adziimaa, S.T, M.Sc.
NPP. 1991201711052

***DIII INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018***

**PERANCANGAN SISTEM PENGENDALIAN LAJU ALIRAN
PADA ALAT KEBOCORAN PIPA SEBAGAI PENUNJANG
PRAKTIKUM SISTEM INSTRUMENTASI INDUSTRI**

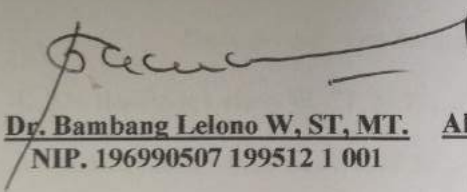
TUGAS AKHIR

Oleh :

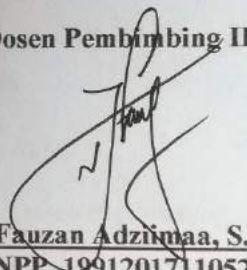
**Shaski Icraini Diotama
NRP. 10 51 15 00000 053**

**Surabaya, 20 Juli 2018
Mengetahui dan Menyetujui,**

Dosen Pembimbing I,


Dr. Bambang Lelono W, ST, MT.
NIP. 196990507 199512 1 001

Dosen Pembimbing II,


Ahmad Fauzan Adzima, S.T, MSc.
NPP. 1991201711052

Kepala Departemen Teknik Instrumentasi,


Dr. Ir. Purwadi Agus Darwito, M.Sc
NIP. 19620822 198803 1 001

**PERANCANGAN SISTEM PENGENDALIAN LAJU
ALIRAN PADA ALAT UJI KEBOCORAN PIPA SEBAGAI
PENUNJANG PRAKTIKUM SISTEM INSTRUMENTASI
INDUSTRI**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
Pada
Program Studi DIII Teknik Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

**Oleh :
Shaski Icraini Diotama
NRP. 10 51 15 00000 053**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Bambang Lelono W, ST, M.T..... (Dosen Pembimbing I)
2. Ahmad Fauzan Adziimaa ST, M.Sc (Dosen Pembimbing II)
3. Dwi Oktavianto Wahyu N., ST, M.T (Dosen Penguji)

SURABAYA
JULI 2018

PERANCANGAN SISTEM PENGENDALIAN LAJU ALIRAN PADA ALAT UJI KEBOCORAN PIPA SEBAGAI PENUNJANG PRAKTIKUM SISTEM INSTRUMENTASI INDUSTRI

Nama : Shaski Icraini Diotama
NRP : 10 51 15 000 053
Departemen : Teknik Instrumentasi
Pembimbing I : Dr. Bambang Lelono W, ST, M.T.
Pembimbing II : Ahmad Fauzan Adziimaa ST, M.Sc.

ABSTRAK

Agar penggunaannya pipa dapat dipergunakan seefisien mungkin dan menghindari terjadinya kegagalan dalam suatu proses maka perlu dilakukan pengujian pipa untuk mengetahui kekuatan pipa. Pada alat uji kebocoran pipa ini terdapat *pressure tank* sebagai penampung fluida bertekanan yang sebagai bahan pengujian bocor pipa. Agar ketersediaan air pada *pressure tank* tercukupi maka diperlukan pengendalian laju aliran. Dengan melakukan pengendalian laju aliran, maka akan dapat mengontrol kecepatan air yang masuk ke *pressure tank*. Pada perancangan sistem pengendalian laju aliran ini, sensor yang digunakan untuk mengukur laju aliran adalah sensor *water flowmeter*. Output dari sensor *water flowmeter* akan mengukur laju aliran yang ada dan akan memberikan respon ke Atmega128 saat *setpoint* laju aliran tercapai sehingga MOV sebagai aktuator akan merespon. Dari data respon pengendalian yang didapat menunjukkan bahwa pengendalian telah berjalan sesuai dengan *setpoint* yang diinginkan yaitu 16 LPM dan beresilasi mendekati *setpoint*. Selain itu berdasarkan pengujian alat pada sistem pengendalian laju aliran didapatkan respon time yang cukup baik untuk mencapai steady state dengan maksimum *overshoot* 18,44 LPM. Waktu yang dibutuhkan untuk mencapai *setpoint* (*settling time*) dari awal pengambilan data adalah ± 45 detik.

Kata Kunci : Pengujian Kebocoran Pipa, Pengendalian Laju Aliran.

**DESIGN OF FLOW CONTROL SYSTEM IN PIPING LEAK
TEST AS A PRACTICAL SUPPORT OF
INSTRUMENTATION INDUSTRY SYSTEM**

Name : Shaski Icraini Diotama
NRP : 10 51 15 000 053
Departement : Instrumentation Engineering
Supervisor I : Dr. Bambang Lelono W, ST, M.T.
Supervisor II : Ahmad Fauzan Adziimaa ST, M.Sc.

ABSTRACT

In order to use, the pipe can be used as effficiently as possible and avoid the occurance of failure in a process it is necessary to test the pipe to determine the strength of a pipe. In this leak pipe test apparatus there is a pressure tank as a container of pressurized fluid which is the material for testing leak pipe. In order for the availability of water at a sufficient pressure tank it is necessary to control flow rate. By controlling the flow rate, it will be able to control the speed of water coming into the pressure tank. In designing this flow control system, the sensor used to measure flow is a water flow meter sensor G3/4". The output of the water flowmeter sensor will measure the current flow and will respond to Atmega128 when the setpoint flow is reached so that the MOV as actuator flow will respond. From the control response data obtained shows the desired changes in accordance with the desired setpoint of 16 LPM and oscillate close to the setpoint. In addition, based on testing at a good enough time to achieve steady state with a maximum overshoot of 18,44 LPM. The time required to reach the setpoint (settling time) from the initial data retrieval is ± 45 seconds.

Kata Kunci : Leak Piping Test, Flow Control System

KATA PENGANTAR

Assalamualaikum. Wr. Wb.

Alhamdulillahirobbil ‘Alamin. Puji syukur kehadiran Allah SWT, atas berkat rahmat dan hidayahnya penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul **PERANCANGAN SISTEM PENGENDALIAN LAJU ALIRAN PADA ALAT UJI KEBOCORAN PIPA SEBAGAI PENUNJANG PRAKTIKUM SISTEM INSTRUMENTASI INDUSTRI**. Shalawat serta salam semoga senantiasa terlimpah curahkan kepada Nabi Muhammad SAW. Sebagaimana telah kita ketahui bersama bahwa dengan rahmat-Nya kita masih diberi kesempatan untuk beribadah kepada-Nya serta masih ada kesempatan bagi kita untuk berpikir jernih untuk masa depan kita. Penulis dapat mengerjakan laporan ini juga tidak lepas dari nikmat yang diberikan oleh-Nya kepada penulis sehingga dapat menyelesaikan laporan ini.

Penulisan laporan ini diajukan untuk memenuhi salah satu syarat kelulusan untuk menyelesaikan program Diploma-III Teknik Instrumentasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya. Banyak dorongan dan bantuan yang penulis dapatkan selama menyelesaikan Tugas Akhir ini sehingga terselesaikannya dengan beberapa kekurangan dan kelebihan. Dalam penyusunan laporan ini, penulis mengucapkan terima kasih atas perhatian dan segala bantuan yang telah diberikan oleh :

- a. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc sebagai Ketua Departemen Teknik Instrumentasi, Fakultas Vokasi, ITS sekaligus sebagai Dosen Wali penulis.
- b. Bapak Dr. Bambang Lelono W, ST, M.Sc. dan Bapak Ahmad Fauzan Adziimaa ST, M.Sc. selaku dosen pembimbing yang senantiasa memberikan motivasi, bimbingan dan arahan dalam menyelesaikan Tugas Akhir.
- c. Bapak dan Ibu dosen Departemen Teknik Instrumentasi yang telah memberikan ilmu selama kuliah.
- d. Mama dan Ayah yang senantiasa mendoakan dan memberikan semangat kepada penulis. Mbak Anggi dan

- Darren yang selalu menjadi penyemangat penulis serta seluruh keluarga besar penulis yang telah membantu serta memberi dukungan baik dalam segi moral maupun materi.
- e. Teman-teman satu kelompok AOTP yang telah berjuang bersama dalam pengerjaan Tugas Akhir ini.
 - f. Teman - teman Teknik Instrumentasi 2015 yang selalu mendukung, membantu, dan memberi semangat saat pengerjaan tugas akhir ini.
 - g. Teman-teman Teknik Fisika 2015 yang selalu memberi semangat kepada teman-teman F50.3.
 - h. Sahabat saya Nadiyah, Amanda, dan Ambar yang menjadi tempat curhat dan penyemangat penulis.
 - i. Semua pihak yang telah membantu sehingga penulis dapat menyelesaikan laporan tugas akhir ini.

Penulis menyadari dalam laporan ini masih terdapat banyak kekurangan di mana-mana, baik di dalam pembahasan materi ataupun dalam penyajian laporan ini. Untuk itu penulis mohon maaf. Dan untuk perbaikan dalam penulisan berikutnya, kami sebagai penulis menerima segala macam kritik, saran, dan pendapat yang bersifat membangun agar dalam penyusunan berikutnya dapat lebih baik dan lebih sempurna. Semoga laporan Tugas Akhir ini dapat berguna bagi penulis dan para pembaca dalam memperluas wawasan dan pengetahuan.

Wassalamualaikum. Wr. Wb.

Surabaya, 20 Juli 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3

BAB II TINJAUAN PUSTAKA

2.1 Sistem Pengendalian Proses	5
2.2 Laju Aliran (<i>Flow</i>).....	6
2.3 <i>Flowmeter</i>	8
2.4 Rotameter	9
2.5 Pompa	11
2.6 <i>Stepper Motor Controlled Valve</i>	11
2.7 Pengendalian Driver Motor Stepper pada MOV	14
2.8 Mikrokontroler ATmega	16
2.9 CodeVision AVR.....	16
2.10 Karakteristik Statik.....	17
2.11 Teori Ketidakpastian.....	18

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Alat dan Bahan	21
3.2 Prosedur Perancangan dan Pembuatan Alat	21
3.2.1 Studi Literatur.....	22
3.2.2 Perancangan Sistem.....	23
3.3 Pembuatan Alat	27
3.3.1 Pembuatan Software	27

3.3.2 Pembuatan Hardware.....	28
3.4 Pengujian Alat dan Analisis	29
3.5 Pembuatan Laporan.....	29

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Rancang Bangun Alat	31
4.2 Data Karakteristik Statik Sensor.....	32
4.3 Kalibrasi Aktuator	35
4.4 Pengujian Alat	36
4.5 Pengendalian Flow dengan Motorized Operated Valve (MOV)	39
4.6 Pembahasan	39

BAB V PENUTUP

5.1 Kesimpulan.....	41
5.2 Saran	41

DAFTAR PUSTAKA

LAMPIRAN A (LISTING PROGRAM)

LAMPIRAN B (DATASHEET ATMEGA128)

LAMPIRAN C (DATASHEET SENSOR)

LAMPIRAN D (MANUAL BOOK POMPA)

LAMPIRAN E (PERHITUNGAN KALIBRASI SENSOR)

DAFTAR GAMBAR

Gambar 2.1 <i>Flow Control Loop</i>	5
Gambar 2.2 Aliran Laminer	7
Gambar 2.3 Aliran Transisi	7
Gambar 2.4 Aliran Turbulen	8
Gambar 2.5 Water Flowmeter	9
Gambar 2.6 Konfigurasi Dasar Rotameter	9
Gambar 2.7 Pompa	11
Gambar 2.8 <i>Stepper Motor Controlled Valve</i>	12
Gambar 2.9 Motor stepper dengan lilitan unipolar	14
Gambar 2.10 Motor stepper empat phase (unipolar).....	14
Gambar 2.11 Logika Pengendalian Motor Stepper	15
Gambar 2.12 Mikrokontroler ATmega128	16
Gambar 3.1 <i>Flowchart</i>	22
Gambar 3.2 Diagram Blok Sistem	23
Gambar 3.3 P&ID	23
Gambar 3.4 Diagram Blok Rangkaian <i>Driver Motor</i> <i>Stepper</i>	24
Gambar 3.5 Rangkaian LCD 16x2	25
Gambar 3.6 Pemasangan Mekanik Alat	26
Gambar 3.7 <i>Minimum Flow/Recycle Line</i>	26
Gambar 3.8 Pedoman Menentukan nilai <i>Minimum Flow</i> ...	27
Gambar 3.9 Tampilan Listing Program Software pada CodeVision AVR	28
Gambar 3.10 Desain 2D Sistem	28
Gambar 4.1 Diagram blok sistem.....	31
Gambar 4.2 Plan Alat Uji Kebocoran Pipa	31
Gambar 4.3 Grafik Histerisis Laju Aliran	33
Gambar 4.4 Grafik Respon Aktuator	38
Gambar 4.5 Grafik Respon Sistem.....	38

DAFTAR TABEL

Tabel 2.1 Tabel Kebenaran Pengendalian Motor Stepper .	15
Tabel 3.1 Spesifikasi Alat	21
Tabel 4.1 Pengujian Karakteristik Sensor	32
Tabel 4.2 Hasil Pengujian Sistem	36

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan industri saat ini sudah berkembang dengan pesat, baik itu berupa industri gas dan pengilangan minyak, industri air minum, pabrik yang memproduksi bahan kimia serta obat-obatan, dan penyediaan energi listrik bagi manusia melalui pembangkit. Dalam menjalankan keberlangsungan industri tersebut sangat dibutuhkan suatu alat pengontrol dalam pengoperasian bahan olahannya [1].

Hampir semua proses industri dalam menjalankan proses produksinya membutuhkan bantuan sistem pengendali, Ada banyak pengendalian yang harus dikendalikan di dalam suatu proses. Diantaranya yang paling umum, adalah tekanan (*pressure*) didalam sebuah *vessel* atau pipa, aliran (*flow*) didalam pipa, suhu (*temperature*) di unit proses seperti *mixing process*, atau permukaan zat cair (*level*) disebuah tangki. Setiap proses selalu mempunyai “keadaan yang diinginkan” dan suatu “keadaan yang berlangsung”. Agar keadaan yang diinginkan sama dengan keadaan yang berlangsung, maka pengontrolan harus dilakukan[2].

Pada alat uji kebocoran pipa ini terdapat pressure tank sebagai penampung fluida bertekanan (air) yang dimana air tersebut sebagai bahan untuk pengujian kebocoran pipa. Agar ketersediaan air pada pressure tank tercukupi maka diperlukan pengendalian laju aliran. Dengan melakukan pengendalian laju aliran, maka akan dapat mengontrol kecepatan air yang masuk ke pressure tank. Apabila tidak dilakukan pengendalian, maka laju aliran yang masuk ke pressure tank dapat tidak terkendali, yang nantinya akan memberikan dampak buruk bagi sistem[3].

Kegiatan proses yang berjalan di industri tidak terlepas dari peralatan–peralatan yang memberlangsungkan proses tersebut, diantaranya unsur perpipaan yang berfungsi sebagai media transportasi (pengaliran) fluida tersebut agar dapat berjalan sesuai dengan proses yang diharapkan. Pipa merupakan saluran tertutup yang biasanya berpenampang lingkaran dan digunakan untuk mengalirkan fluida dengan tampang aliran penuh. Fluida yang

dialirkan melalui pipa bisa berupa zat cair atau gas, tekanan bisa lebih besar atau lebih kecil dari tekanan atmosfer. Sebagaimana kita ketahui penggunaan pipa sudah begitu luas baik dikalangan masyarakat maupun pada industri skala kecil, menengah, maupun skala besar, yang berfungsi untuk menyalurkan cairan dari satu tempat ke tempat lainnya. Agar dalam penggunaannya pipa dapat dipergunakan seefisien mungkin dan menghindari terjadinya kegagalan dalam suatu proses akibat kerusakan pada pipa maka perlulah dilakukan pengujian pipa untuk mengetahui kekuatan suatu pipa. Pengujian kebocoran pipa dimaksudkan untuk dapat dijadikan pegangan bagi penyelenggara pembangunan dalam mengawasi dan memeriksa mutu pipa[4].

Dari beberapa tinjauan tersebut, maka dalam Tugas Akhir ini dilakukan perancangan sistem pengendalian laju aliran pada alat uji kebocoran pipa sebagai penunjang praktikum sistem instrumentasi industri. Sistem kontrol atau pengendalian yang dimaksudkan adalah pengendalian laju aliran pada *storage tank* untuk menjaga volume air pada *pressure tank* agar sesuai dengan set point yaitu dengan mengatur debit alirannya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah :

1. Bagaimana merancang sistem pengendalian laju aliran pada alat uji kebocoran pipa sebagai penunjang praktikum sistem instrumentasi industri?
2. Bagaimana cara mendapatkan debit yang sesuai dengan set point agar volume air pada *pressure tank* terpenuhi?

1.3 Tujuan

Adapun tujuan dari tugas akhir ini adalah :

1. Untuk membangun sistem pengendalian laju aliran pada alat uji kebocoran pipa sebagai penunjang praktikum sistem instrumentasi industri.
2. Untuk mengetahui besarnya laju aliran yang sesuai dengan set point berdasarkan dari kebutuhan *pressure tank* yang diperlukan.

1.4 Batasan Masalah

Adapun batas ruang lingkup dari penelitian tugas akhir ini sebagai berikut ;

1. Variabel yang dikontrol adalah laju aliran yang masuk ke dalam *pressure tank*.
2. Variabel yang dimanipulasi adalah laju aliran yang melalui MOV.

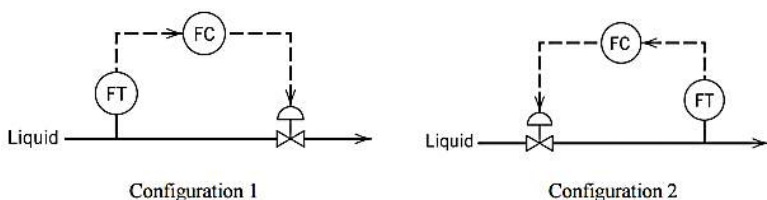
Halaman ini sengaja dikosongkan

BAB II TINJAUAN PUSTAKA

2.1 Sistem Pengendalian Proses

Hampir semua proses di industri membutuhkan peralatan-peralatan otomatis untuk mengendalikan parameter-parameter prosesnya. Otomatisasi tidak hanya diperlukan demi kelancaran operasi, keamanan, ekonomi maupun mutu produk, tetapi lebih merupakan kebutuhan pokok. Suatu proses industri tidak akan dapat dijalankan tanpa bantuan sistem pengendalian. Parameter – parameter yang umum dikendalikan dalam suatu proses adalah tekanan (*pressure*), laju aliran (*flow*), suhu (*temperature*) dan permukaan zat cair (*level*). Gabungan serta alat – alat pengendalian otomatis itulah yang disebut sistem pengendalian proses (*process control system*). Sedangkan semua peralatan yang membentuk sistem pengendalian tersebut disebut instrumentasi pengendalian proses (*process control instrumentation*).

Secara garis besar suatu rangkaian pengendalian proses dibagi menjadi 4 langkah, yaitu mengukur – membandingkan – menghitung – mengkoreksi. Langkah pertama yaitu mengukur, mengukur merupakan tugas dari sensor. Langkah berikutnya adalah membandingkan apakah hasil pengukuran dari sensor sudah sesuai dengan apa yang dikehendaki. Apabila terjadi ketidaksesuaian antara *set point* dengan hasil pengukuran maka akan dilakukan perhitungan untuk menentukan aksi apa yang dilakukan supaya sesuai dengan set point yang diinginkan. Pada langkah kedua dan ketiga ini adalah tugas dari pengendali. Langkah terakhir adalah melakukan pengkoreksian yang merupakan tugas dari actuator[5].



Gambar 2.1 Flow Control Loop [5].

2.2 Laju Aliran (*Flow*)

Debit atau laju aliran merupakan suatu koefisien yang menyatakan jumlah volume air yang mengalir dari suatu sumber dalam satuan waktu tertentu melalui suatu penampang air, sungai, saluran, pipa atau kran dan sebagainya, biasanya diukur dalam satuan liter/detik atau liter/menit. Dalam pengukuran fluida termasuk penentuan tekanan, kecepatan, debit, gradien kecepatan, turbulensi dan viskositas. Terdapat banyak cara melaksanakan pengukuran-pengukuran, misalnya langsung, tak langsung, gravimetrik, volumetrik, elektronik, elektromagnetik dan optik. Pengukuran debit secara langsung terdiri dari atas penentuan volume atau berat fluida yang melalui suatu penampang dalam suatu selang waktu tertentu. Metoda tak langsung bagi pengukuran debit memerlukan penentuan tinggi tekanan, perbedaan tekanan atau kecepatan di beberapa titik pada suatu penampang dan dengan besaran perhitungan debit. Metode pengukuran aliran yang paling teliti adalah penentuan gravimetrik atau penentuan volumetrik dengan berat atau volume diukur atau penentuan dengan mempergunakan tangki yang dikalibrasikan untuk selang waktu yang diukur. Pengukuran laju aliran merupakan salah satu hal yang sangat penting dalam proses plant boiler. Pengukuran debit air ditujukan untuk mengetahui kecepatan aliran pada satuan waktu[6]. Untuk mengetahui debit maka harus mengetahui satuan ukuran volume dan satuan ukuran waktu terlebih dahulu, karena debit berkaitan dengan satuan volume dan satuan waktu. Untuk menentukan debit air menggunakan persamaan:

$$Q = \frac{V}{t} \quad (2.1)$$

dimana:

Q = debit (liter/s)

V = volume (liter)

t = waktu (s)

Secara garis besar jenis aliran dapat dibedakan atau dikelompokkan sebagai berikut:

a. Aliran Tunak (*steady*)

Suatu aliran dimana kecepatannya tidak terpengaruh oleh perubahan waktu sehingga kecepatan konstan pada setiap titik (tidak mempunyai percepatan).

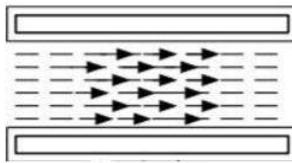
b. Aliran Tidak Tunak (*unsteady*)

Suatu aliran dimana terjadi perubahan kecepatan terhadap waktu.

Adapun macam aliran fluida dapat diaktegorikan sebagai berikut:

a. Aliran Laminar

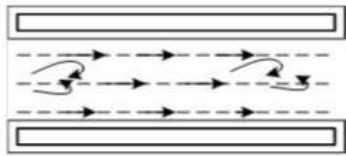
Aliran laminar didefinisikan sebagai aliran dengan fluida yang bergerak dalam lapisan–lapisan atau lamina–lamina dengan satu lapisan meluncur secara lancar. Aliran laminar ini mempunyai nilai bilangan Reynoldsnnya kurang dari 2300 ($Re < 2300$).



Gambar 2.2 Aliran Laminer.[6]

b. Aliran Transisi

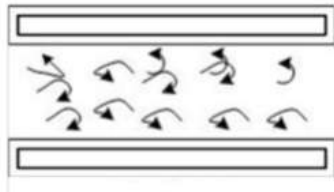
Aliran transisi merupakan aliran peralihan dari aliran laminar ke aliran turbulen. Keadaan peralihan ini tergantung pada viskositas fluida, kecepatan dan lain-lain yang menyangkut geometri aliran dimana nilai bilangan Reynoldsnnya antara 2300 sampai dengan 4000 ($2300 < Re < 4000$).



Gambar 2.3 Aliran Transisi.[6]

c. Aliran Turbulen

Aliran turbulen didefinisikan sebagai aliran yang dimana pergerakan dari partikel-partikel fluida sangat tidak menentu karena mengalami percampuran serta putaran partikel antar lapisan, yang mengakibatkan saling tukar momentum dari satu bagian fluida ke bagian fluida yang lain dalam skala yang besar. Dimana nilai bilangan Renoldsnya lebih besar dari 4000 ($Re > 4000$).



Gambar 2.4 Aliran Turbulen.[6]

2.3 Flowmeter

Water Flowmeter FS400a ini mempunyai ukuran drat 3/4", sistem kerja sama dengan water flow meter sensor ukuran lain yaitu bekerja sesuai prinsip induksi dari eek Hall berkaitan dengan deteksi fluks magnetik. Air yang mnegalir dilewatkan pada sebuah rotor yang berputar dengan laju angular sesuai dengan debit air yang mengalir. Pada ujung roto terdapat kepingan magnet alam yang ditempatkan sedemikian rupa ditengah kumparan kawat. Akibat perputaran keping magnet terhadap kumparan ini tercipta medan magnet, yang dideteksi oleh detektor *hall effect* dan dikonversi menjadi arus listrik. Semakin cepat magnet berputar akibat debit air yang membesar , semakin besar dinamo pada pembangkit listrik yang dihasilkan (prinsipnya seperti dinamo pada pembangkit listrik tenaga air) yang dapat dibaca oleh mikrokontroller yang tehubung dengan sensor ini. Water flowmeter ini mampu mendeteksi debit air hingga 60 liter per menit. Spesifikasi G 3/4 inch water flow sensor: [17]

- Rentang tegangan operasional : 5-24 VDC
- Maksimum arus operasional : 15 mA (pada 5VDC)
- Maksimum suhu air : $\leq 120^{\circ}\text{C}$
- Maksimum suhu operasional : $\leq 80^{\circ}\text{C}$

- Maksimum tekanan air : ≤ 2 MPa



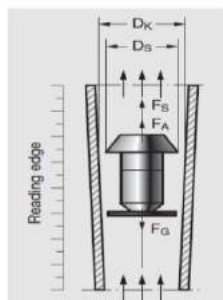
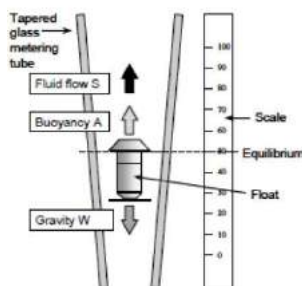
Gambar 2.5 Water Flowmeter [17].

Dimana output dari water flowmeter ini yaitu frekuensi. Persamaan untuk merubah frekuensi ke debit yaitu :

$$F(\text{Hz}) = 5,4 \times Q(\text{Lmin})$$

2.4 Rotameter

Rotameter adalah sebuah alat ukur laju aliran tipe variabel area. Pada alat ini, terdapat sebuah *float* yang mengambang yang berada di dalam tabung berbentuk meruncing tegak, *float* diangkat ke posisi keseimbangan diantara gaya ke bawah dan gaya keatas dari fluida sebagai tambahan efek *buoyancy* dari fluida mengalir bertemu *float* melalui annular *orifice*. Laju aliran dapat dibaca dengan melihat posisi *float* tersebut. Fluida mengalir melalui tabung dari bawah ke atas, mengangkat *float* keatas. Diameter tabung bertambah kearah atas float naik ke titik dimana gaya keatas *float* terbentuk oleh perbedaan tekanan pada jarak *annular*, diantara *float* dan tabung, sama dengan berat *float*[7].



Gambar 2.6 Konfigurasi Dasar Rotameter [7].

Gambar 2.6. menunjukkan 3 gaya yang bekerja pada *float*, yaitu:

1. Gaya gravitasi konstan, W
2. *Bouyancy* A , berdasarkan prinsip Archimedes, konstan jika densitas fluida konstan
3. Gaya S , gaya keatas fluida yang melewati *float*

Persamaan laju aliran pada variabel area flowmeter sebagai berikut :

$$Q = CA \sqrt{\rho} \quad (2.2)$$

Dimana :

Q = Aliran

C = Konstanta yang bergantung pada *float*

A = Luas penampang yang dilalui fluida

ρ = Densitas fluida

Rotameter biasanya memiliki akurasi hingga 2% dari skala penuh dan kemampuan pengulangannya (*repeatability*) mencapai 0,25% dari pembacaannya. Mereka memiliki rentang akurasi hingga 10:1, yang mana lebih baik dari pada flowmeter yang pembacaannya bergantung pada akar kuadrat dari *pressure drop*. Kalibrasi dari rotameter berlaku hanya pada satu fluida yang densitasnya khusus. Rotameter dapat digunakan untuk mengukur laju aliran dari gas atau liquid. Kegunaannya terbatas untuk laju aliran yang cukup kecil [7].

2.5 Pompa

Pompa adalah suatu alat atau mesin untuk memindahkan cairan dari satu tempat ketempat lain melalui suatu media perpipaan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus. Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian masuk (*suction*) dengan bagian keluar (*discharge*). Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (*penggerak*) menjadi tenaga kinetis (*kecepatan*), dimana tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran.[8]



Gambar 2.7 Pompa.[8]

Pengubah energi mekanik menjadi energi tekan fluida tersebut di atas dapat dicapai dengan beberapa cara, antara lain :

1. Mengubah energi mekanis dengan menggunakan alat semacam sudut atau impeler dengan bentuk tertentu.
2. Dengan menggunakan gerak bolak –balik piston atau alat semacamnya.
3. Dengan penukaran energi menggunakan fluida perantara, baik gas atau cair. Fluida perantara ini diberi kecepatan tinggi dan dicampur dengan fluida yang dipompa yang berkecepatan rendah. Cara ini biasa digunakan pada pompa jet (jet pump).
4. Dengan menggunakan udara atau gas bertekanan tinggi yang diinjeksikan ke dalam suatu saluran yang berisi fluida yang dipompa. Cara ini digunakan pada air/gas lift pump.

2.6 Stepper Motor Controlled Valve

Salah satu jenis *motor control valve* dc yang dikendalikan dengan pulsa-pulsa digital. *Prinsip* kerja motor stepper adalah bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit dimana motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor stepper tersebut.[9]



Gambar 2.8 Stepper motor Controlled Valve.[9]

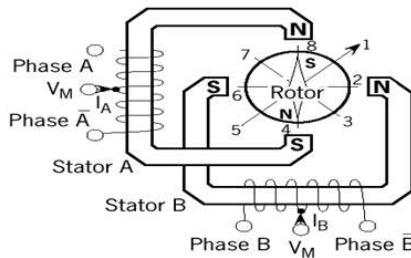
Stepping motor memiliki beberapa keunggulan-keunggulan sebagai berikut:

- a. Mudah dikendalikan dengan putaran sinkron yang hanya tergantung pada denyut yang diterima.
- b. Memiliki sudut yang tepat pada setiap langkah tanpa adanya akumulasi kesalahan sudut.
- c. Memiliki respon memulai dan berhenti yang baik pada saat digunakan, dan dapat digunakan sebagai servo motor.
- d. Putaran sinkron pada kecepatan yang sangat rendah dapat dicapai selagi “*carriage*” dihubungkan langsung pada shaft *stepping motor*.
- e. Memiliki ketahanan yang tinggi karena masa pengoperasiannya tergantung pada ketahanan “*bearing*” dan tidak pada bagian yang menempel semisal sikat atau “brush”.
- f. Berbagai macam sudut dapat dicapai tergantung dari tipe motor.
- g. Memiliki torsi menahan di posisi yang diinginkan pada saat motor dalam keadaan diam.
- h. Tidak terdapat sisa arus yang mengalir meskipun putaran terhalang oleh kelebihan beban.
- i. Kecepatan putaran tidak dipengaruhi oleh pergantian beban, karenanya kecepatan dapat selalu terjaga secara simultan dengan denyut masuk.

Motor stepper disini berfungsi sebagai penggerak valve secara elektrik. Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodic.^[10] Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. Keunggulannya antara lain adalah :

- Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
- Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
- Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
- Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
- Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC
- Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya
- Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

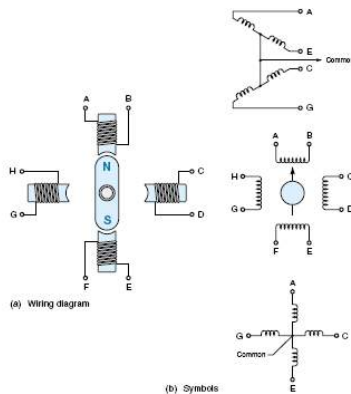
Berdasarkan metode perancangan rangkain pengendalinya, motor stepper dapat dibagi menjadi jenis unipolar dan bipolar. Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (ground) pada salah satu terminal lilitan (wound) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (VM) pada bagian tengah (center tap) dari lilitan.



Gambar 2.9 Motor stepper dengan lilitan unipolar.[10]

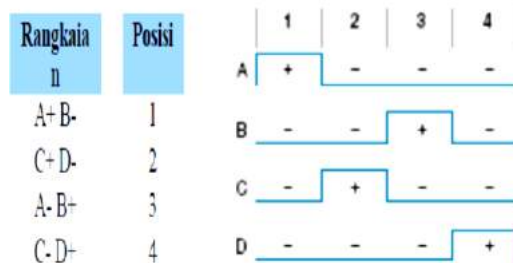
2.7 Pengendalian Driver Motor Stepper pada MOV

Apabila motor stepper dihubungkan dengan piranti digital atau I/O port, maka dibutuhkan rangkaian *interface*. Hal ini sangat penting karena jumlah arus yang diperlukan untuk memberikan energi pada pasangan-pasangan kumparan lebih besar dari kemampuan I/O port, sehingga dibutuhkan sejumlah rangkaian penyangga (buffer) yang akan menguatkan arus untuk dapat menggerakkan motor stepper. Rangkaian driver MOV menggunakan rangkaian modul driver L298 yang mampu bekerja hingga arus 4A, sedangkan beban motor yang digunakan yaitu 3A. Berikut ini adalah ilustrasi struktur motor stepper sederhana dan pulsa yang dibutuhkan untuk menggerakkannya.^[11]



Gambar 2.10 Motor stepper empat phase (unipolar).[11]

Motor stepper empat phase memiliki empat kumparan medan yang dapat diberikan energi secara terpisah/tersendiri, dan arus pada motor stepper unipolar selalu mengalir dalam arah yang sama melalui kumparan. Cara seerhana untuk mengoperasikan motor stepper empat-phase adalah dengan memberikan energi phase satu pada suatu waktu yang berurutan (dikenal dengan *wave drive*). Untuk memutar ke arah CW (clockwise, searah jarum jam) step yang dilakukan dengan memberikan step pada motor ini adalah dengan mmeberikan energi secara bergantian pada AB kemudian CD untuk menarik rotor dari kutub ke kutub.[11]



Gambar 2.11 Logika Pengendalian Motor Stepper.[11]

Tabel 2.1 Tabel kebenaran pengendalian motor stepper

Kumparan Searah Jarum Jam (Close Valve)				Kumparan Berlawanan Jarum Jam (Open Valve)			
1	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0
1	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0
dst				dst			

2.8 Mikrokontroler ATmega

Mikrokontroler merupakan keseluruhan sistem komputer yang dikemas menjadi sebuah *chip* di mana di dalamnya sudah terdapat Mikroprosesor, I/O, Memori bahkan ADC, berbeda dengan Mikroprosesor yang berfungsi sebagai pemroses data. Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam 1 siklus *clock* atau dikenal dengan teknologi RISC (*Reduced Instruction Set Computing*). Secara umum, AVR dapat dikelompokkan ke dalam 4 kelas, yaitu keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Pada dasarnya yang membedakan masing-masing adalah kapasitas memori, *peripheral* dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama [12]. Berikut ini gambar Mikrokontroler Atmega128 .



Gambar 2.12 Mikrokontroler ATmega128 [12].

2.9 CodeVision AVR

CodeVision AVR merupakan sebuah *software* yang digunakan untuk memprogram mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya diprogram ke dalam mikrokontroler menggunakan fasilitas yang sudah disediakan oleh program tersebut. Salah satu *compiler* program yang umum digunakan sekarang ini adalah CodeVision AVR yang

menggunakan bahasa pemrograman C. CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya *codewizard*, fasilitas ini memudahkan pengguna atau pemrogram dalam inisialisasi mikrokontroler yang akan digunakan [13].

2.10 Karakteristik Statik

Karakteristik statik pengukuran merupakan karakteristik yang ditentukan melalui perhitungan matematik atau secara grafik. Karakteristik statik merupakan karakter yang menggambarkan parameter dari sebuah instrument pada saat keadaan *steady* [14]. Karakteristik statik terdiri dari:

a. *Range*

Range merupakan selisih antara nilai minimum dan maksimum yang terukur oleh suatu instrument atau alat ukur.

b. Akurasi

Akurasi merupakan tingkat ketelitian suatu alat dalam memberikan hasil pengukuran.

c. Toleransi

Toleransi menunjukkan kesalahan maksimum yang diperbolehkan pada hasil pembacaan alat ukur.

d. Sensitivitas

Sensitivitas menunjukkan perubahan output instrument yang terjadi saat diberi kualitas pengukuran yang berbeda. Sensitivitas diperoleh dari $\Delta O / \Delta I$, dimana O adalah output dan I adalah input.

e. *Repeatability*

Repeatability merupakan kemampuan *instrument* dalam menampilkan ulang output pengukuran yang sama pada pengukuran yang berulang.

f. Linearitas

Linearitas merupakan grafik yang menampilkan pengukuran nilai sebenarnya (input) yang dapat menghasilkan output nilai yang ditunuukkan oleh instrument (output). Pengukuran yang baik adalah ketika inut pengukuran dan output pengukuran berbanding lurus (linear). Linearitas dapat diperoleh dari persamaan berikut:

$$O-O_{\min} = \frac{O_{\max}-O_{\min}}{I_{\max}-I_{\min}} (I-I_{\min}) \quad (2.3)$$

Persamaan Linearitas :

$$O_{\text{ideal}} = KI+a \quad (2.4)$$

Dimana:

$$K = \frac{O_{\max}-O_{\min}}{I_{\max}-I_{\min}} \quad (2.5)$$

Dan

$$a = O_{\min}-KI_{\min} \quad (2.6)$$

2.11 Teori Ketidakpastian

Ketidakpastian pengukuran merupakan tingkat seberapa besar ketidakpastian yang dihasilkan oleh suatu alat ukur [15]. Dalam menghitung ketidakpastian pengukuran ada beberapa langkah yang harus dihitung, antara lain:

a. Koreksi

Koreksi dapat diperoleh dengan persamaan berikut

Koreksi = Pembacaan standard-Pembacaan alat

b. Standard deviasi

$$\sigma = \frac{\sqrt{\Sigma(Di - D')^2}}{n-1} \quad (2.7)$$

dimana :

Di = koreksi alat ukur

Di' = rata-rata koreksi

n = Banyak range pengukuran

c. Analisa Tipe A, (Ua)

Pada analisa tipe A ini hasilnya diperoleh dari data pengukuran. Adapun persamaannya adalah sebagai berikut:

$$Ua1 = \frac{\sigma_{maks}}{\sqrt{n}} \quad (2.8)$$

$$Ua2 = \sqrt{\frac{SSR}{n-2}} \quad (2.9)$$

d. Analisa Type B, (U_b)

Analisa tipe B ini diperoleh berdasarkan sertifikat kalibrasi atau spesifikasi dari alat ukur. Adapun persamaannya adalah sebagai berikut:

$$U_{b1} = \frac{\text{Resolusi}/2}{\sqrt{3}} \quad (2.10)$$

Dimana :

$SSR = \text{Sum Square Residual}$

$$U_{b2} = \frac{a}{k} \quad (2.11)$$

e. Ketidakpastian Kombinasi (U_c)

U_c merupakan Ketidakpastian kombinasi dari ketidakpastian tipe A dan ketidakpastian tipe B. Adapun persamaan dari ketidakpastian kombinasi adalah:

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2} \quad (2.12)$$

Halaman ini sengaja dikosongkan

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1 Alat dan Bahan

Adapun alat dan bahan yang digunakan untuk mengerjakan tugas akhir ini adalah sebagai berikut :

1. Pompa Shimizu PS-135 E
2. Water Flow Sensor
3. Rotameter
4. Mikrokontroler ATMEGA 128
5. MOV (Motor Operation Valve)
6. Driver Motor Stepper L298N
7. Pipa PVC uji sebanyak 5 buah dengan panjang 25 cm

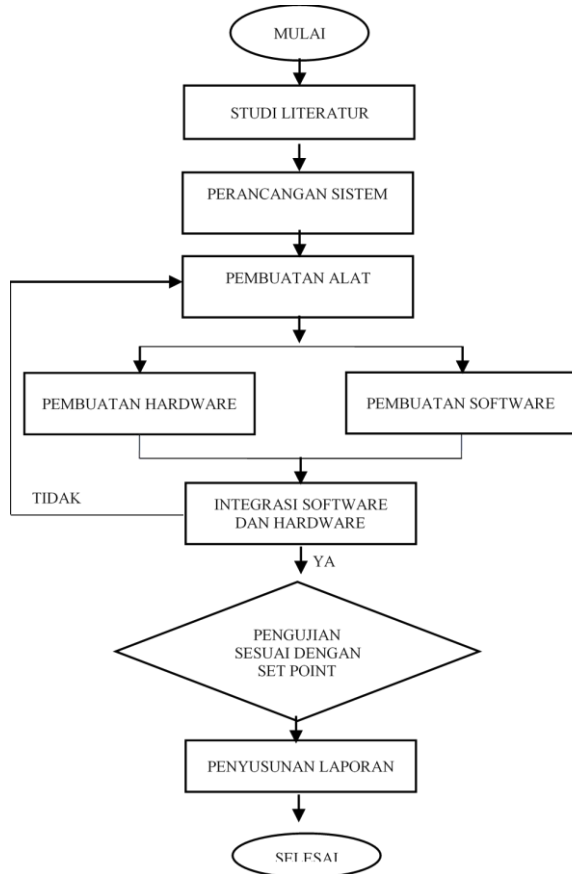
Berikut merupakan spesifikasi dari masing-masing alat :

Tabel 3.1 Spesifikasi Alat

Komponen	Spesifikasi
Pompa	Debit pompa = 28 L/m
<i>Pressure Tank</i>	Volume = 50 L
<i>Storage Tank</i>	Volume = 60 L
MOV (Motor Stepper dan Gate Valve)	Dengan sudut putar 360° dan valve dengan diameter 3/4"
Water Flow Sensor	Range = 1 - 60 LPM
Rotameter	Range = 0 - 36 LPM

3.2 Prosedur Perancangan dan Pembuatan Alat

Langkah-langkah perancangan alat ini digambarkan dalam *flowchart* penelitian yang dapat dilihat pada gambar berikut ini :



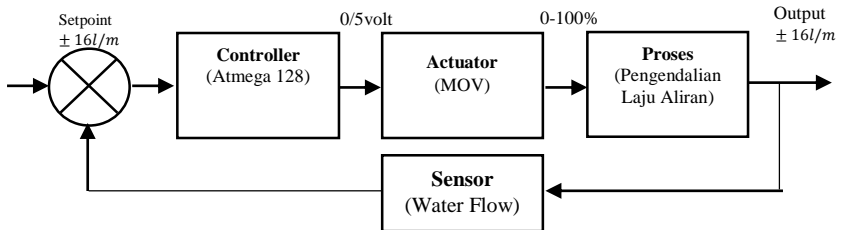
Gambar 3.1 *Flowchart*

3.2.1. Studi Literatur

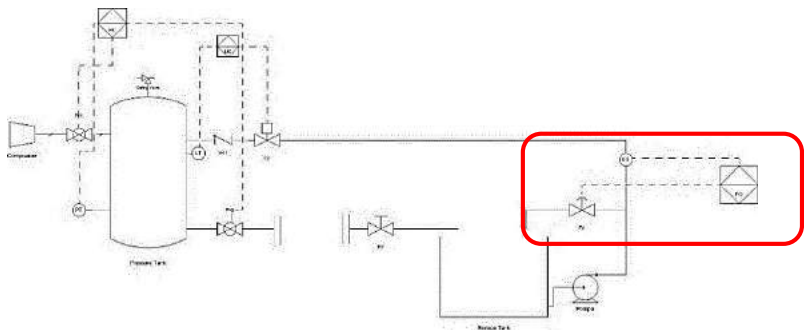
Kegiatan ini dilakukan dengan mencari materi penunjang dari buku, jurnal dan artikel pada media cetak maupun online. Mencari dasar teori yang tepat dalam merancang alat, sistem kerja dari sistem pengendalian *flow feed water* pada alat uji kebocoran pipa sebagai penunjang praktikum sistem instrumentasi industri. Serta melakukan studi literatur yang berhubungan dengan sistem pengendalian menggunakan Atmega 128.

3.2.2. Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem pengendalian laju aliran pada alat uji kebocoran pipa. Perancangan sistem akan menjelaskan tentang perancangan-perancangan elektrik dan mekanik alat. Berikut adalah Diagram blok serta P&ID dari pengendalian laju aliran pada alat uji kebocoran pipa.



Gambar 3.2 Diagram Blok Sistem



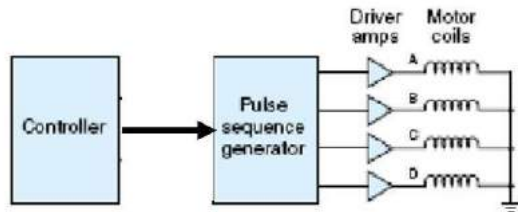
Gambar 3.3 P&ID

Gambar 3.2 merupakan diagram blok sistem pengendalian *flow feed water* pada alat uji kebocoran pipa, mikrokontroler Atmega128 sebagai *controller*. *Valve* yang di *couple* (MOV) sebagai aktuator, sedangkan alat uji kebocoran pipa sebagai tempat proses. Sensor yang digunakan pada pengendalian *flow* tersebut adalah *flow sensor*. *Setpoint* (SP) merupakan besar proses variabel yang dikehendaki. *Error* merupakan proses mengurangi besaran *set point* dengan pengukuran *process variable*, *Manipulated Variable* (MV) merupakan input dari

suatu proses yang dimanipulasi agar besarnya variabel proses sama dengan set blok diatas menjelaskan mengenai alur pengendalian yang dilakukan oleh sensor. Mulai dari input, apabila sensor mendeteksi adanya error ketika laju aliran fluida tidak sesuai dengan set point, sinyal akan diproses oleh controller yang dapat menggerakkan actuator. Actuator inilah yang akan mengatur laju aliran agar dapat sesuai dengan set point. Set point debit laju aliran pada tugas akhir ini sebesar $\pm 16 \text{ L/m}$.

- **Perancangan Driver Motor Stepper**

Perancangan motor stepper dibutuhkan rangkaian driver motor stepper untuk memberikan energi(energizing) pada pasangan-pasangan kumparan lebih besar dari kemampuan I/O port, terdapat dimana rangkaian ini berisi 4 rangkaian yang diparalel. Setiap 1 rangkaian dihubungkan ke masing – masing kutub kumparan pada motor stepper. Motor stepper pada plant ini dipasang pada gate valve yang berguna sebagai aktuator pengendalian flow.



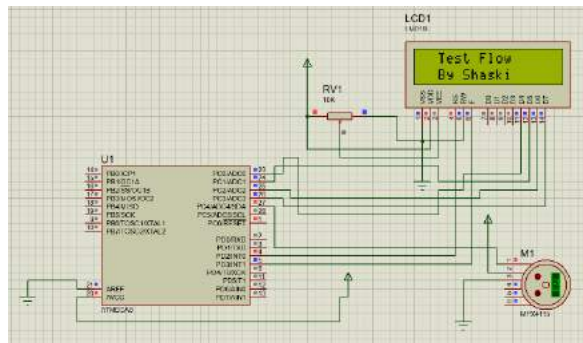
Gambar 3.4 Diagram blok rangkaian driver motor stepper [11].

Gambar 3.4 ini pada dasarnya hanya merupakan rangkaian switching arus yang mengalir lilitan pada motor stepper. Urutan pemberian data pada motor stepper ini dapat mengontrol arah putaran dari motor stepper ini. Penambahan kecepatan pada motor stepper dapat dilakukan dengan cara meningkatkan frekuensi pemberian data pada rangkaian switching arus. Rangkaian driver motor stepper ini nantinya terhubung langsung dengan lilitan pada motor, rangkaian power suplai, dan rangkaian

mikrokontroler yang pada akhirnya menentukan kapan lilitan yang diinginkan dalam kondisi off atau on.

- Perancangan Tampilan LCD 16x2

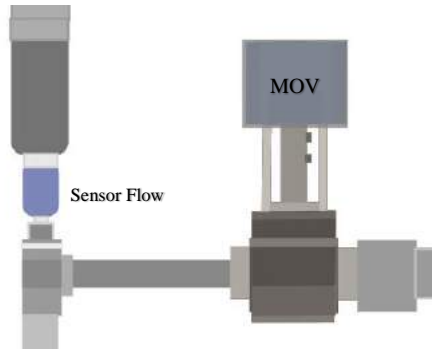
Modul LCD (*Liquid Crystal Display*) yang digunakan dalam tugas akhir ini adalah modul LCD yang dapat menampilkan 16x2 karakter, yaitu 16 karakter untuk kolom dan 2 karakter untuk baris. *Display* LCD adalah suatu *modul* penampil. Dalam hal ini LCD digunakan sebagai tampilan masukkan *set point* selanjutnya digunakan sebagai tampilan nilai *set point* dan nilai laju aliran yang terbaca oleh *water flow sensor*.



Gambar 3.5 Rangkaian LCD 16x2

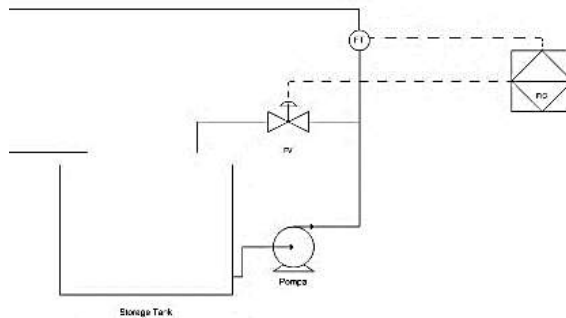
- Perancangan Peletakkan Mekanik Alat

Dalam pemasangan flowmeter dan MOV ini tidak boleh dilakukan secara sembarangan. Terdapat standar yang harus dilakukan dalam pemasangannya yaitu jarak pemasangan antar instrument ini. Hal ini sangat penting untuk dilakukan agar sistem pengendalian aliran dapat berjalan dengan baik dan terhindar dari turbulensi. Adapun pemasangan flowmeter dan MOV pada plan sesuai dengan gambar dibawah.



Gambar 3.6 Pemasangan Mekanik Alat

- Perancangan *Minimum Flow / Recycle Line*
 Fungsi dari *recycle line* adalah menjaga agar jumlah aliran yang dibutuhkan pompa selalu berada dalam kondisi aman jika terjadi sesuatu pada *line* distribusi yang menyebabkan fluida tidak bisa dialirkan ke dalam jalur distribusi. Prinsip kerja *recycle line* yaitu satu line menuju proses dan satu line menuju *storage tank*. Pada saat line proses hanya membutuhkan sedikit fluida, secara otomatis MOV akan membuka lebih besar untuk mengalirkan kembali fluidanya ke *storage tank* dan begitu sebaliknya.



Gambar 3.7 *Minimum Flow/Recycle Line*

Untuk menentukan berapa nilai *minimum flow* nya, dapat dilihat gambar dibawah ini.

Type of Service	Recommended Minimum Flow, % of Design	NPSH _r /NPSH _a
Hydrocarbons, single-and double-suction	25-50	1.1
Water and water solutions, single-suction	40-70	2.0
Water and water solution, double-suction*	70-85	2.0
Pumps with domed head/flow curves	Below shutoff head	1.1-2.0

* Generally above 1,000 gpm.

Gambar 3.8 Pedoman untuk menentukan nilai *minimum flow* [16].

Dari gambar diatas, diketahui nilai *minimum flow* untuk proses ini adalah 40% dari desain pompa. Dimana pada sistem ini menggunakan pompa dengan debit 28 LPM.

Y (nilai *minimum flow*)

$$Y = 40\% \times 28 \text{ LPM}$$

$$Y = 11,2 \text{ LPM}$$

Maka setpoint laju aliran pada sistem ini adalah debit pompa dikurangi dengan nilai minimum flow yaitu sebesar $\pm 16 \text{ LPM}$.

3.3 Pembuatan Alat

Pada tahap pembuatan alat ini akan dibagi menjadi 2 bagian yaitu pembuatan hardware alat dan pembuatan software alat. Untuk pembuatan hardware terdiri dari pembuatan elektrik dan mekanik pada alat, sedangkan pembuatan software berisi langkah-langkah cara mengkodeing mikrokontroller agar sistem control dapat berjalan sesuai dengan yang diinginkan.

3.3.1. Pembuatan Software

Pada tahap ini perancangan *software* merupakan siklus gabungan untuk rangkaian pemrosesan sinyal. Rangkaian pemrosesan sinyal pada alat ini menggunakan mikrokontroler Atmega128. Untuk membuat program pada Atmega128, dibutuhkan *software* CodeVision AVR. Pada program ini, akan diberikan *listing program* untuk mengolah sinyal masukan. Sinyal masukan dari *sensing element*.


```

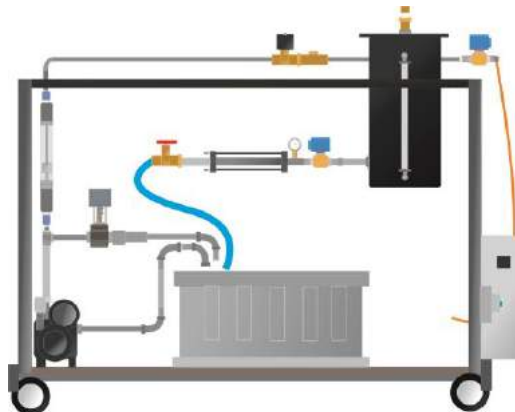
1  /*****
2  This program was produced by the
3  CodeWizardAVR V2.05.3 Standard
4  Automatic Program Generator
5  © Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
6  http://www.hpinfotech.com
7
8  Project :
9  Version :
10 Date   : 23/06/2018
11 Author : USER
12 Company :
13 Comments:
14
15
16 Chip type           : ATmega128
17 Program type        : Application
18 AVR Core Clock frequency: 16.000000 MHz
19 Memory model        : Small
20 External RAM size    : 0
21 Data Stack size     : 1024
22 *****/
23
24 #include <mega128.h>
25
26 // Atmega128 LCD functions

```

Gambar 3.9 Tampilan Listing Program Software pada CodeVision AVR

3.3.2. Pembuatan Hardware

Pembuatan hardware meliputi pembuatan sistem pada plan alat uji kebocoran pipa.



Gambar 3.10 Desain 2D Sistem

3.4 Pengujian Alat dan Analisis

Tahapan ini adalah tahapan terpenting dari metode pelaksanaan lainnya, dikarenakan tahapan ini merupakan tolok ukur keberhasilan perancangan sistem pengendalian laju aliran pada alat uji kebocoran pipa sebagai penunjang praktikum sistem instrumentasi industri. Pengujian dilakukan diantaranya untuk mengetahui:

- a. Tingkat keberhasilan alat dalam mengendalikan proses.
- b. Tingkat keberhasilan sistem dalam menguji kebocoran pipa.
- c. Dapat karakteristik statik hasil uji dari respon sistem.

3.5 Pembuatan Laporan

Pembuatan laporan dilakukan setelah seluruh tahap terselesaikan sehingga hasil yang diperoleh dari pembuatan alat dapat dijelaskan secara rinci dan spesifik sesuai dengan data-data yang diperoleh.

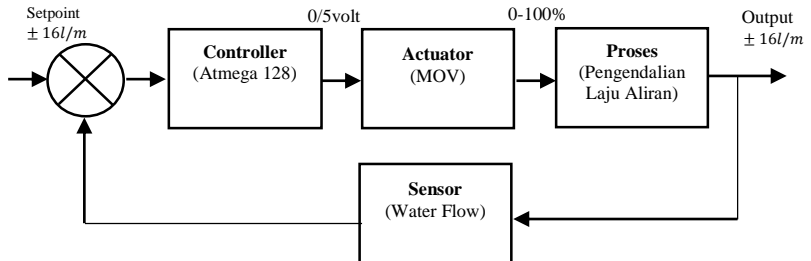
Halaman ini sengaja dikosongkan

BAB IV

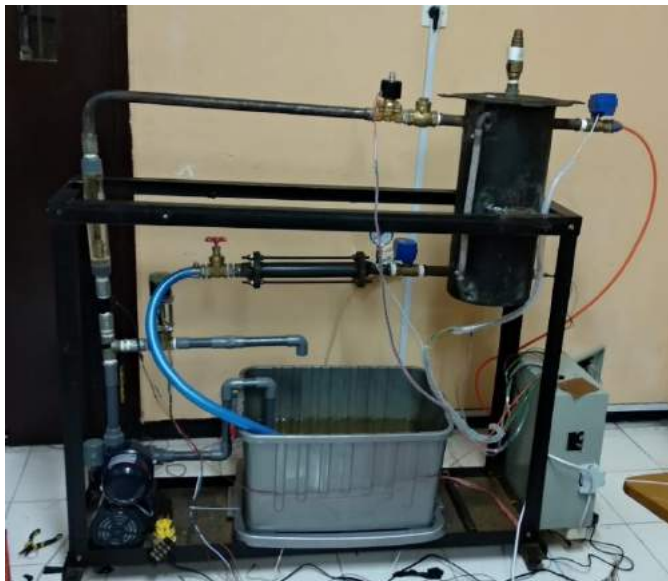
HASIL DAN PEMBAHASAN

4.1 Hasil Rancang Bangun Alat

Berikut ini adalah perancangan sistem pengendalian laju aliran pada alat uji kebocoran pipa sebagai penunjang praktikum sistem instrumentasi industri.



Gambar 4.1 Diagram blok sistem



Gambar 4.2 Plan Alat Uji Kebocoran Pipa

Pada proses ini menggunakan sensor flowmeter dengan actuator berupa motor stepper yang dapat mengendalikan aliran pada *storage tank* sebelum memasuki *pressure tank* dengan setpoint laju aliran sebesar $\pm 16\text{L/min}$. Berdasarkan blok diagram diatas, sensor akan mendeteksi laju aliran (*flow*) dan nilai flownya dijaga agar sesuai dengan set point. Jika nilai *flow* pada *sensor* kurang dari set point maka *controller* (mikrokontroller ATmega128) mengirimkan sinyal ke aktuator (MOV) agar membuka hingga tercapainya nilai *set point*.

4.2 Data Karakteristik Statik Sensor

Karakteristik statik merupakan sifat dari sebuah alat ukur yang tidak bergantung terhadap waktu. Berikut merupakan data perhitungan. Untuk mengetahui nilai untuk mengetahui karakteristik suatu alat.

Tabel 4.1 Pengujian Karakteristik Sensor

No.	Pembacaan STD (xi)	Pembacaan Sensor		Rata2 Pembacaan Alat	Histerisis
		Rata Naik	Rata Turun		
1	3	3.69	3.73	3.71	0.04
2	5	5.53	5.71	5.62	0.18
3	7	7.38	7.47	7.425	0.09
4	9	9.22	9.35	9.285	0.13
5	11	11.07	11.23	11.15	0.16
JMLH	35	36.89	37.49	37.19	0.6
RATA2	7	7.378	7.498	7.438	0.12

Berikut ini hasil perhitungan nilai karakteristik statik berdasarkan data pada tabel 4.2

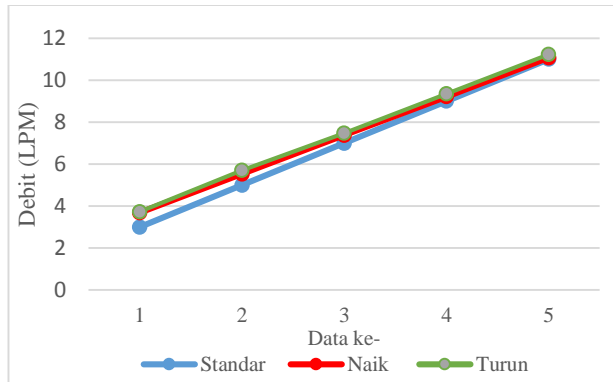
- Histerisis

$H(I) = O(I)_I \uparrow - O(I)_I \downarrow$; $\hat{H} = H(I)_{max}$; sehingga maksimum histerisis didapat sebesar

$$\% \text{maks histerisis} = \frac{\hat{H}}{O_{max} - O_{min}} \times 100\%$$

$$= \frac{0,18}{11,07-3,69} \times 100 \% = 2,43 \%$$

Berikut Gambar 4.3 merupakan grafik Histerisis dari hasil pengukuran laju aliran naik dan turun.



Gambar 4.3 Grafik Histerisis Laju Aliran

- Sensitivitas : $\frac{\Delta O}{\Delta I} = \frac{11,15 - 3,71}{11 - 3} = 0,93$
- Akurasi (A) = $1 - \left| \frac{y_n - x_n}{y_n} \right| \times 100\% = 98 \%$
- Error (e) = $1 - A = 1 - 0.98176 = 0,0182$
= 1,82%

Dari data yang terdapat dalam tabel maka dapat diketahui nilai karakteristik alat sebagai berikut:

- Range: 3,69 LPM - 11,07 LPM
- Span: 7,38 LPM
- Resolusi: 0,01 cm
- Histerisis: 2,43 %
- Sensitivitas : 0,93
- Akurasi : 98%
- Error : 1,82%

Berikut ini merupakan cara untuk mencari nilai ketidakpastian *sensor water flow*.

- a. Nilai Ketidakpastian Tipe A

$$\sigma = \frac{\sqrt{\sum(y-\bar{y})^2}}{n-1} \dots\dots\dots(4.1)$$

$$\sigma = 0,231100627$$

Sehingga didapatkan nilai U_{a1} sebagai berikut:

$$U_{a1} = \frac{\sigma}{\sqrt{n}} \dots\dots\dots(4.2)$$

$$U_{a1} = \frac{0,2311}{\sqrt{5}} = 0,10335$$

Sedangkan nilai ketidakpastian regresi $U_{a2} = \sqrt{\frac{SSR}{n-2}}$

Dimana:

SSR (*Sum Square Residual*)= $\sum SR$ (*Square Residual*)

$SR = R^2$ (*Residu*)

Y = Nilai koreksi

$$Y_{reg} = a + (b \cdot xi) \dots\dots\dots(4.3)$$

$$a = \bar{y}_l + (b \cdot \bar{x}) \dots\dots\dots(4.4)$$

$$b = \frac{n \cdot \sum xy - \sum x \sum y}{n \cdot \sum x^2 - (\sum x)^2} \dots\dots\dots(4.5)$$

$$SSR = \sum (y - Y_{reg})^2 \dots\dots\dots(4.6)$$

Dimana:

x = *Pemb. standar*,

y = *Nilai koreksi*, n = *Jumlah data*

b = -0,03139

Sehingga nilai:

$$a = (-0,438) + (-0,03139 \times 7)$$

$$a = -0,21821$$

Jadi, persamaan regresi menjadi:

$$Y_{reg} = (-0,21821) + (x(-0,03139))$$

Yang menghasilkan nilai $SSR = 0,435$

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} = \sqrt{\frac{0,435}{5-2}} = 0,3811$$

- b. Nilai Ketidakpastian *Type B* :

Pada tipe ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi (U_{b1}) dan Ketidakpastian alat standar rotameter (U_{b2}). Dengan perhitungan sebagai berikut:

$$U_{b1} = \frac{\frac{1}{2}xResolusi}{\sqrt{3}} = \frac{\frac{1}{2}x0,01}{\sqrt{3}} = 0,002886$$

$U_{b2} = \frac{a}{k}$, dikarenakan pada alat standar tidak ada sertifikat kalibrasinya maka nilai a (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0. Sehingga hasil : $U_{b2} = 0$

- c. Nilai ketidakpastian kombinasi U_c :

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2} \dots\dots\dots (4.7)$$

$$U_c = 0.39491$$

4.3 Kalibrasi Aktuator

Aktuator yang digunakan pada sistem ini yaitu motor stepper dengan valve tipe gate valve, dimana kondisi dari pembukaan 0% hingga pembukaan 100%, yaitu 4,5 kali putaran 360°. Spesifikasi motor stepper yaitu 1,8°/step. Maka step yang diperlukan untuk membuka valve 100% adalah sebagai berikut:

- **Pembukaan 100% valve**

$$\text{Pembukaan 100\% valve} = 4,5 \times 360^\circ$$

$$= 1620^\circ$$
- **Step yang diperlukan untuk pembukaan 100% valve**

$$\text{Step} = 1620^\circ : 1,8^\circ$$

$$= 900 \text{ step pulsa}$$

4.4 Pengujian Alat

Pengujian alat dilakukan dengan mencari data pengujian dari *input* dan *output* alat sebagai nilai pengonversi dari nilai keluaran sensor menjadi nilai yang sesuai dengan besaran fisis yang diukur. Pengujian sensor dilakukan dengan memberikan flowrate menggunakan pompa air. Hasil pembacaan sensor dapat dilihat seperti pada tabel 4.1. pengambilan data dilakukan setiap 5 detik. Actuator motor stepper dapat memberikan output mendekati setpoint sensor sebesar $\pm 16\text{L/min}$. Pengujian dilakukan untuk mengetahui grafik respon actuator dan pembacaan flowrate pada sensor berdasarkan waktu yang telah ditentukan.

Tabel 4.2 Hasil Pengujian Sistem

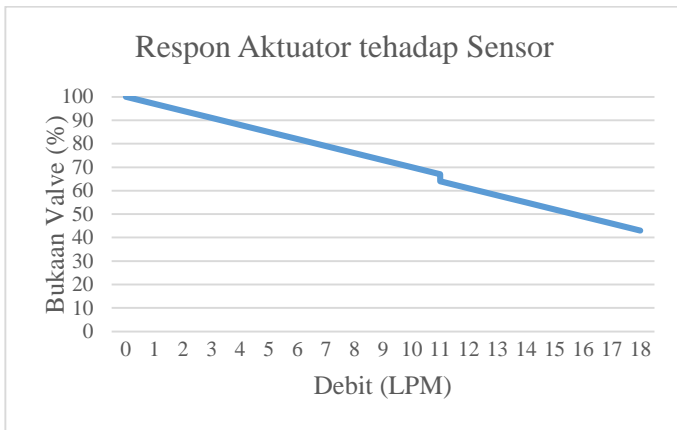
Menit ke-	Detik ke-	Debit	Buka Valve	Menit ke-	Detik ke-	Debit	Buka Valve
1	1	0,92	100	1	46	16,6	49
	2	0,92	100		47	17,52	46
	3	0,92	100		48	17,52	46
	4	0,92	100		49	18,44	43
	5	1,84	97		50	18,44	43
	6	1,84	97		51	16,6	49
	7	1,84	97		52	16,6	49
	8	2,76	94		53	15,68	52
	9	2,76	94		54	15,68	52
	10	3,69	91		55	16,6	49
	11	3,69	91		56	16,6	49
	12	4,61	88		57	15,68	52
	13	4,61	88		58	15,68	52
	14	4,61	88		59	16,6	49
	15	5,53	85		60	16,6	49
	16	5,53	85	2	61	15,68	52
	17	5,53	85		62	15,68	52
	18	6,45	82		63	16,6	49
	19	6,45	82		64	16,6	49
	20	7,38	79		65	15,68	52

Tabel 4.2 Hasil Pengujian Sistem

Menit ke-	Detik ke-	Debit	Buka Valve	Menit ke-	Detik ke-	Debit	Buka Valve
1	21	7,38	79	2	66	15,68	52
	22	8,3	76		67	16,6	49
	23	8,3	76		68	16,6	49
	24	8,3	76		69	15,68	52
	25	9,22	73		70	15,68	52
	26	9,22	73		71	16,6	49
	27	10,14	70		72	16,6	49
	28	10,14	70		73	15,68	52
	29	10,14	70		74	15,68	52
	30	11,07	67		75	16,6	49
	31	11,07	67		76	16,6	49
	32	11,07	67		77	15,68	52
	33	11,99	64		78	15,68	52
	34	11,99	64		79	16,6	49
	35	12,91	61		80	16,6	49
	36	12,91	61		81	15,68	52
	37	12,91	61		82	15,68	52
	38	13,83	58		83	16,6	49
	39	13,83	58		84	16,6	49
	40	14,76	55		85	15,68	52
	41	14,76	55		86	15,68	52
	42	14,76	55		87	16,6	49
	43	15,68	52		88	16,6	49
	44	15,68	52		89	15,68	52
	45	16,6	49		90	15,68	52

Berdasarkan data diatas, didapatkan yaitu respon actuator terhadap sensor dan respon sistem terhadap waktu.

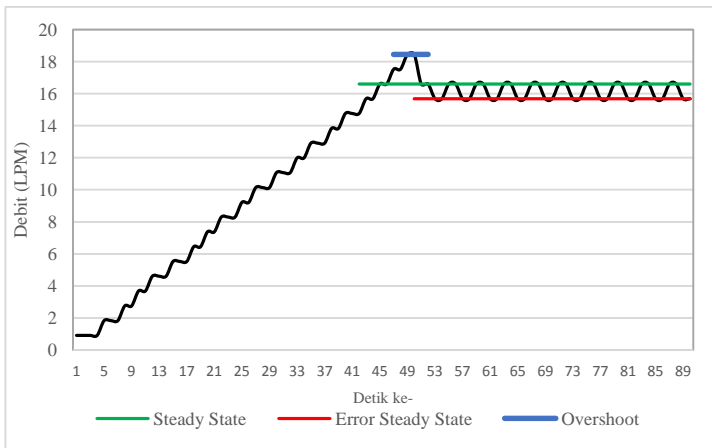
- Respon Aktuator terhadap Sensor



Gambar 4.4 Grafik Respon Aktuator

Pada data respon actuator dapat diketahui bahwa pada sistem ini apabila debit kurang dari setpoint yaitu sebesar 16 LPM maka actuator akan menutup sampai debit terpenuhi.

- Respon Sistem terhadap Waktu



Gambar 4.5 Grafik Respon sistem

Dari grafik diatas adalah data flowrate yang di deteksi oleh sensor dengan setpoint ± 16 LPM dan dapat diketahui *maximum overshoot* dari sistem yaitu 18,44 LPM dengan error steady state sebesar 14,76 LPM dan steady state sebesar 16,6 LPM. Dari hasil pengujian diketahui waktu yang diperlukan MOV bergerak hingga setpoint flow tercapai (*rise time*) yaitu 45 detik. Aliran yang di kendalikan oleh motor stepper dapat bergerak dengan menyesuaikan setpoint yang telah ditentukan.

4.5 Pengendalian Flow dengan Motorized Operated Valve (MOV)

Pengendalian Flow yaitu dengan metode mengendalikan pembukaan valve pada output pompa agar output pompa sesuai dengan setpoint yang diinginkan. Dimana setpoint output pompa yang diinginkan yaitu 16 LPM sedangkan output pompa yaitu 28 LPM. Adapun metode yang digunakan dalam pengendalian flow adalah sebagai berikut :

- Pada saat plant diaktifkan, maka sensor level low pada pressure tank akan aktif karena tanki masih kosong.
- Saat aktif, maka MOV akan bergerak membuka dan kemudian pompa menyala.
- Pada saat pompa aktif, maka flowmeter yang berada di bagian output pompa akan mengatur berapa flow yang melewatinya.
- Gerakan membuka valve akan berhenti ketika setpoint flow telah tercapai.
- Valve akan mempertahankan posisi pembukaannya hingga sensor level high aktif dan solenoid akan aktif.
- Flowmeter akan mensensing debit sebesar 0 LPM maka kemudian MOV akan bergerak membuka dan melakukan *recycle line* aliran.

4.6 Pembahasan

Pada tugas akhir ini telah dirancang sisitem pengendalian laju aliran pada alat uji kebocoran pipa. Sistem pengendalian alairan ini bertujuan untuk mengetahui besarnya laju aliran yang sesuai dengan set point berdasarkan dari kebutuhan *pressure tank*

yang diperlukan. Pengujian sensor dilakukan untuk mnertahui karakteristik dari sensor tersebut. Dari pengujian tersebut dapat diketahui bahwa sensor memiliki karakteristik statik histerisis sebesar 2,43%, keakurasian sensor sebesar 98%, dan error sebesar 1,82%. Pengujian pengendalian flow dengan menggunakan aktuator MOV, MOV pada sistem ini menggunakan motor jenis stepper dan gate valve. Flowmeter yang digunakan yaitu flowmeter ukuran 3/4" dengan range kerja antara 1liter/menit hingga 60 liter/menit. Rangkaian driver MOV menggunakan rangkaian modul driver L298 yang mampu bekerja hingga arus 4A, sedangkan beban motor yang digunakan yaitu 3A. Cara pengendalian MOV yaitu sesuai dengan Tabel 4.5. Dari kalibrasi yang dilakukan, diketahui error flowmeter sebesar 1,82%. Pengendalian aliran yang dilakukan yaitu mengatur bukaan MOV agar output sensor sesuai dengan setpoint yang ditentukan. Setpoint yang diinginkan yaitu ± 16 LPM. Metode pengendalian yang dilakukan yaitu saat pembacaan sensor flow kurang dari 16 LPM, maka MOV akan bergerak menutup dan ketika pembacaan flow lebih dari 16 maka MOV akan bergerak membuka hingga flow mencapai setpoint yang ditentukan yaitu ± 16 LPM. Dari hasil pengujian diketahui waktu yang diperlukan MOV bergerak hingga setpoint flow tercapai (*rise time*) yaitu 45 detik dengan maximum overshoot yaitu 18,44 LPM. Dari proses pengujian tersebut dapat dilihat bahwa proses tersebut memerlukan controller untuk menjamin agar proses tersebut dapat berjalan dengan baik dan mencapai setpoint yang diinginkan.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian laju aliran pada perancangan sistem pengendalian laju aliran pada alat uji kebocoran pipa, dapat disimpulkan, yaitu:

1. Pada pengujian karakteristik alat ukur, sensor water flow memiliki sensitivitas hingga 0,93 LPM dengan presentase akurasi mencapai 98 %.
2. Terjadinya *osilasi* pada pembacaan tegangan pada generator dengan *setpoint* yang ditentukan yaitu ± 16 LPM menggunakan *water flow sensor* dan aktuator berupa *motor stepper*.
3. Aktuator berupa motor stepper berjalan sesuai *setpoint* yang diinginkan yaitu saat pembacaan debit aliran mendekati 16 liter/menit maka *motor stepper* akan berhenti dan valve akan diam. Data pengujian alat pada sistem pengendalian, didapatkan *respon time* yang cukup baik untuk mencapai *steady state* dengan maksimum *overshoot* 18,44 liter/menit dan waktu yang dibutuhkan untuk mencapai *set point (settling time)* dari awal pengambilan data adalah ± 45 detik.

5.2 Saran

Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan, antara lain:

- 1 Untuk menghindari *losse* saat pengendalian, dipastikan bahwa plant sudah tidak ada yang bocor karena akan mempengaruhi pengukuran *flow*
- 2 Untuk menghindari *disturben* perhatikan jarak pemasangan *flowmeter* dari sumber turbulensi agar didapatkan pengukuran *flow* yang stabil.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Saputri, Septriani Dwie. (2009). “Rancang Bangun Venturimeter berbasis Mikrokontroller”. Skripsi Universitas Indonesia. Jakarta
- [2] Wisudana, Dion Hendra. 2012. “Rancang Bangun Sistem Pengendalian Level Unit *Storage Tank A* pada Simulator *Mixing Process*”. Tugas Akhir Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember. Surabaya.
- [3] Divo, Arbi . 2010. “Sistem Kerja Sensor Tabung Venturi untuk Pengukuran Laju Aliran Fluida dalam Pipa di Laboratorium Pabrik PTKI”. Tugas Akhir Universitas Sumatra Utara
- [4] Sarjito, Subroto, Arif Kurniawan. 2016. “ Studi Distribusi Tekanan Aliran melalui Pengecilan Saluran secara Mendadak dengan Belokan pada Penampang Segi Empat”. Jurusan Teknik Mesin, Universitas Muhammadiyah Surakarta. Surakarta.
- [5] Sofyana, Nur Ummama. 2017. “Rancang Bangun Sistem Pengendalian Laju Aliran dan Rekonsiliasi Data pada *Splitting Piping*”. Tugas Akhir Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember. Surabaya.
- [6] Munson, Donald F.Young, Theodore H. Okiishi . 2003. “Mekanika Fluida 1st Edition” . Jakarta : Erlangga.
- [7] Crabtree, Michael A . 2009. Thesis “ Industrial Flow Measurement”. The University of Huddersfield.
- [8] Surindra, M. Denny. 2009. “Pengaruh Putaran Motor Penggerak Pompa Jenis Pompa MMO 65-5 terhadap Efisiensi Pompa dengan Debit 74,3 M³/H saat *Perfomance Test* Produksi Pompa”. Jurusan Teknik Mesin Politeknik Negeri Semarang. Semarang.
- [9] Mukromin, Radian Indra. 2017. “Rancang Bangun Sistem Pengendalian *Flow Rate Inlet* Kolom Purifikasi pada Proses Purifikasi Biogas dengan *Water Scrubber System* berbasis Atmega128”. Tugas Akhir Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember. Surabaya.
- [10] Faisal, Muhammad Reza. 2016. “Rancang Bangun Sistem Pengendalian *Flow* menggunakan *Motorized Operated Valve* (MOV) berbasis PLC pada *Process Control Plant*”. Tugas

Akhir Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember. Surabaya.

- [11] Syahrul. 2015. “Motor Stepper : Teknologi, Metoda dan Rangkaian Kontrol”. Majalah Ilmiah UNIKOM, Universitas Komputer Indonesia. Bandung
- [12] Heryanto M.A dan Adi W.P. 2008. “Pemrograman Bahasa C untuk Mikrokontroller 8535”. Jakarta : Penerbit Andi
- [13] Sidik, Nurcahyo. 2012. “ Aplikasi dan Teknik Pemrograman Mikrokontroller AVR” . Jakarta : Penerbit Andi
- [14] Cahyati ,Yeni. 2012. “Karakteristik Statik Pengukuran”. Pascasarjana Universitas Brawijaya. Malang
- [15] Bentley, John P .2005.. “ *Principles of Measurement System 4th Edition*” . London : Prentice Hall.
- [16] Heald, C.C, dan R. Ralgrave. 1985. “ Backflow Control Improves Pump Performance”. Oil and Gas J., hal 96-105.
- [17] www.seedstudio.com/wiki/G3/4_Water_Flow_sensor

LAMPIRAN A LISITNG PROGRAM

/******

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 30/06/2018
Author :
Company :
Comments:

Chip type : ATmega128
Program type : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 1024

*****/

```
#include <mega128.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
#include <alcd.h>
```

```
#define up PINA.0
#define down PINA.1
#define ok PINA.2
#define valve PORTA.3
#define mov_a1 PORTA.4
#define mov_a2 PORTA.5
#define mov_b1 PORTA.6
#define mov_b2 PORTA.7
```

```

#define b3 PORTE.2
#define b2 PORTE.3
#define b1 PORTE.4
#define b0 PORTE.5
// ----- DEKLARASI SENSOR FLOW ----- //
float freq;
char buff0[30];
int dur,m=0;
int vol,step;
// ----- DEKLARASI SENSOR PRESSURE ----- //
eeprom int sp;
int flag=0,simpan_sp;
float data_mpx;
float tekanan;
int x=43, y=255;
char buff1[30];
char temp[16];
// ----- DEKLARASI SENSOR LEVEL ----- //
int adc,adc2, av_adc2,av_adc,data_av_adc2,data_av_adc, i;
char buff2[30];
char buff5[30];
char buff6[30];
// ----- DEKLARASI RTC ----- //
unsigned char sc,mt,hr;
unsigned char mg,dd,mm,yy;
char buff3[30];
char buff4[30];
// ----- BATAS SUCI DEKLARASI ----- //
#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE

```

```
#define UPE 2
#endif
```

```
#ifndef DOR
#define DOR 3
#endif
```

```
#ifndef FE
#define FE 4
#endif
```

```
#ifndef UDRE
#define UDRE 5
#endif
```

```
#ifndef RXC
#define RXC 7
#endif
```

```
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
#pragma used+
char getchar1(void)
{
    char status,data;
    while (1)
    {
        while (((status=UCSR1A) & RX_COMPLETE)==0);
        data=UDR1;
        if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
            return data;
    }
}
#pragma used-
```

```

// Write a character to the USART1 Transmitter
#pragma used+
void putchar1(char c)
{
while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
UDR1=c;
}
#pragma used-
#define _ALTERNATE_PUTCHAR_
#include <stdio.h>
#define USART0 0           // agar pembacaan tidak acak
#define USART1 1
unsigned char poutput;

void putchar(char c)
{
switch (poutput)
{
case USART0: // the output will be directed to USART0
while ((UCSR0A & DATA_REGISTER_EMPTY)==0);
UDR0=c;
break;
case USART1: // the output will be directed to USART1
while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
UDR1=c;
break;
};
}
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
// m++;
}
#define ADC_VREF_TYPE 0x60
// Read the 8 most significant bits
// of the AD conversion result

```

```

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}
#include <spi.h>
// I2C Bus functions
#include <i2c.h>

#include <twi.h>

// DS1307 Real Time Clock functions
#include <ds1307_twi.h>

// ----- KODING SENSOR + AKTUATOR FLOW ----- //
void baca_flow(){
    TIMSK=0x04;
    TCCR1B=0x07;
    delay_ms(100);
    TCCR1B=0x00;
    TIMSK=0x00;
    dur=TCNT1;
    freq=(((dur+m*65536)*600)/5.4)*0.0166;
    vol=freq;
    TCNT1=0x0000;
    m=0;
}
// STEPPER //
void searah_jj(){
    b0=0; b1=0; b2=0; b3=1;

```

```

    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
}

void berlawanan_jj(){
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
    b0=1; b1=0; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=1; b2=0; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=1; b3=0;
    delay_ms(5);
    b0=0; b1=0; b2=0; b3=1;
    delay_ms(5);
}

void stopped(){
    b3=0; b2=0; b1=0; b0=0
// -----KODING SENSOR + AKTUATOR PRESSURE ----- //

```

```

void baca_tekanan(){
    data_mpx=read_adc(3);
    tekanan=((data_mpx-x)/(y-x))*7;
}
// ----- KODING SENSOR + AKTUATOR LEVEL ----- //
void baca_level_switch_atas(){
    for(i=0;i<20;i++){
        adc=read_adc(0);
        av_adc=av_adc+adc;
        delay_ms(10);
    }
    av_adc=av_adc/20;
    if(av_adc==0){
        data_av_adc=1;
    }
    else{
        data_av_adc=0;
    }
}
void baca_level_switch_bawah(){
    for(i=0;i<20;i++){
        adc2=read_adc(1);
        av_adc2=av_adc2+adc2;
        delay_ms(10);
    }
    av_adc2=av_adc2/20;
    if(av_adc2==0){
        data_av_adc2=1;
    }
    else{
        data_av_adc2=0;
    }
}
// ----- TAMPIL LCD ----- //
void tampil_lcd() {
// FLOW //
lcd_gotoxy(0,1);
lcd_putsf("F:");

```



```

ftoa(freq,2,buff0);
lcd_puts(buff0);
// PRESSURE //
lcd_gotoxy(0,2);
lcd_putsf("P:");
ftoa(tekanan,2,buff1);
lcd_puts(buff1);
// LEVEL //
lcd_gotoxy(0,0);
sprintf(buff2,"LSA:%d LSB:%d",data_av_adc,data_av_adc2);
lcd_puts(buff2);
sprintf(buff5,"%d",data_av_adc);
sprintf(buff6,"%d",data_av_adc2);
}
// ----- RTC ----- //
void get_time()
{
    rtc_get_time(&hr,&mt,&sc);
    rtc_get_date(&mg,&dd,&mm,&yy);
}
void main(void)
{
    // Declare your local variables here
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
    Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T
    State1=T State0=T
    PORTA=0x0F;
    DDRA=0b11111000;
    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
    Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T
    State1=T State0=T
    PORTB=0x00;

```

```
DDRB=0x00;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTC=0x00;
DDRC=0x00;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTD=0x00;
DDRD=0x00;
// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTE=0x00;
DDRE=0xFF;
// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTF=0x00;
DDRF=0x00;
// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=0xFF
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 15,625 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
```

```
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer3 Stopped
// Mode: Normal top=0xFFFF
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
```

```
// INT7: Off
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;
ETIMSK=0x00;
// USART0 initialization
UCSR0A=0x00;
UCSR0B=0x98;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;
// USART1 initialization
UCSR1A=0x00;
UCSR1B=0x98;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x67;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AVCC pin
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
// SPI initialization
// SPI disabled
SPCR=0x00;
// TWI initialization
// TWI disabled
TWCR=0x00;
```

```

twi_master_init(100);
rtc_init(0,0,0);
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(20);
// Global enable interrupts
#asm("sei")
valve=1; //aktif high, buka
mov_a1=1;
mov_a2=0;
mov_b1=1;
mov_b2=0;
delay_ms(1000);
mov_a1=1;
mov_a2=1;
mov_b1=1;
mov_b2=1;
// SET WAKTU //
hr = 22;
mt = 12;
sc = 00;
dd = 18;
mm = 07;
yy = 18;
rtc_set_time(hr,mt,sc);
rtc_set_date(mg,dd,mm,yy);
while (1)

```

```

{
// Place your code here
while(up&&down&&ok){
    baca_level_switch_atas();
    baca_level_switch_bawah();
    baca_tekanan();
    baca_flow();
    tampil_lcd();
    get_time();
    // DATA LOGGER //
    sprintf(buff4, "%02d:%02d:%02d:%02d:%02d ; %s
L/min\n\r",dd,mm,yy,hr,mt,sc,buff0);
    poutput = USART1;
    puts(buff4);
    // HMI INTEGRASI //
    sprintf(buff3,"%s L/min ; %s Bar ;LSA=%s;LSB=%s \r",
buff0,buff1,buff5,buff6);
    poutput = USART0;
    puts(buff3);
    if(vol>0&&vol<16){
        lcd_gotoxy(1,3);
        lcd_puts("mov tutup");
        berlawanan_jj();
    }
    else if(vol>16.5){
        searah_jj();
        lcd_gotoxy(1,3);
        lcd_putsf("mov buka");
    }
    else {
        stopped();
        lcd_gotoxy(1,3);
        lcd_putsf("mov henti");
    }
    if(data_av_adc==1 && data_av_adc2==1){
        valve=0;
    }
    if(tekanan>=simpan_sp){

```

```

        mov_a1=1;
        mov_a2=0;
        mov_b1=0;
        mov_b2=1;
    }
    else if(tekanan<simpan_sp){
        mov_a1=0;
        mov_a2=1;
        mov_b1=1;
        mov_b2=0;
    }
    if(!up){
        delay_ms(50);
        sp++;
        sprintf(temp,"%d ",sp);
        lcd_gotoxy(12,0);
        lcd_puts(temp);
    }
    else if(!down){
        delay_ms(50);
        sp--;
        sprintf(temp,"%d ",sp);
        lcd_gotoxy(12,0);
        lcd_puts(temp);
    }
    else if(!ok){
        delay_ms(50);
        simpan_sp=sp;
        flag=1;
        lcd_clear();
        delay_ms(1000);
        lcd_gotoxy(0,0);
        sprintf(temp,"%d",simpan_sp);
        lcd_puts(temp);
        delay_ms(1000);
        lcd_clear();
    }
}

```


LAMPIRAN B

(DATA SHEET SYSTEM ATMEGA128)

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 133 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 128Kbytes of In-System Self-programmable Flash program memory
 - 4Kbytes EEPROM
 - 4Kbytes Internal SRAM
 - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Up to 64Kbytes Optional External Memory Space
 - Programming Lock for Software Security
 - SPI Interface for In-System Programming
- QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Two 8-bit PWM Channels
 - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
 - Output Compare Modulator
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Dual Programmable Serial USARTs
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
 - Software Selectable Clock Frequency
 - ATmega103 Compatibility Mode Selected by a Fuse
 - Global Pull-up Disable
- I/O and Packages
 - 53 Programmable I/O Lines
 - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V ATmega128L
 - 4.5 - 5.5V ATmega128
- Speed Grades
 - 0 - 8MHz ATmega128L
 - 0 - 16MHz ATmega128



**8-bit Atmel
Microcontroller
with 128KBytes
In-System
Programmable
Flash**

**ATmega128
ATmega128L**

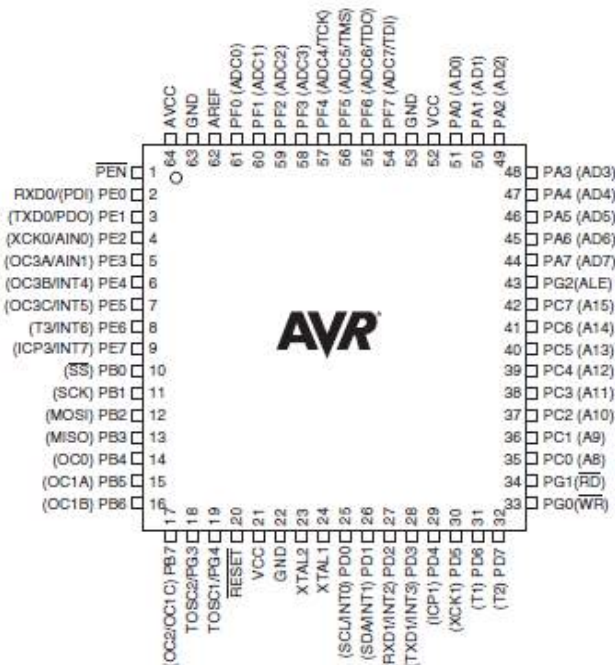
Summary

Rev. 246705-AV14-00/11



Pin Configurations

Figure 1. Pinout ATmega128



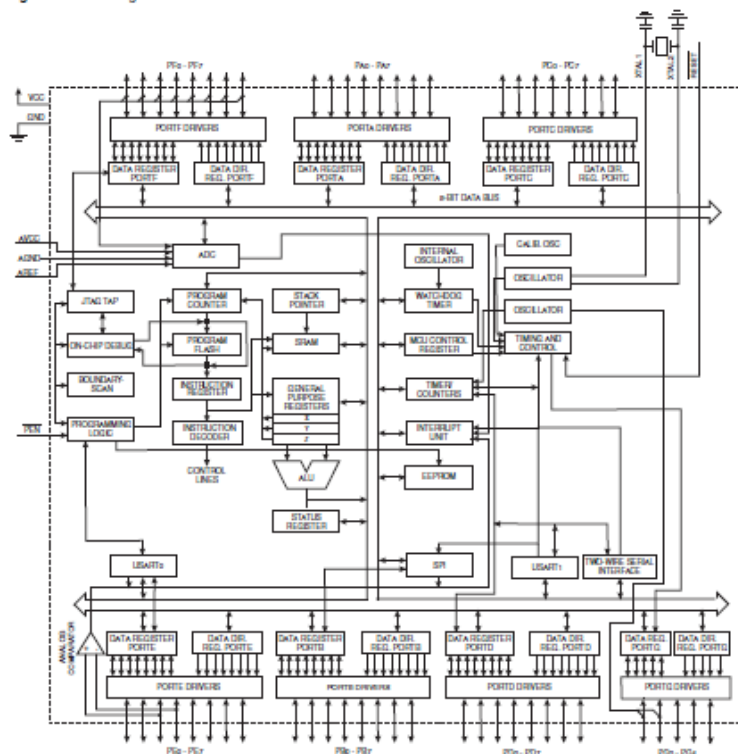
Note: The Pinout figure applies to both TOFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

ATmega103 Compatibility Mode

By programming the M103C fuse, the Atmel® ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, Asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16-bit Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port C is output only.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.

In addition, there are some other minor differences to make it more compatible to ATmega103:

- Only EXTRF and PORF exists in MCUCSR.
- Timed sequence not required for Watchdog Time-out change.
- External Interrupt pins 3 - 0 serve as level interrupt only.
- USART has no FIFO buffer, so data overrun comes earlier.

Unused I/O bits in ATmega103 should be written to 0 to ensure same operation in ATmega128.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega128 as listed on [page 72](#).

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega128 as listed on [page 73](#).

Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port C also serves the functions of special features of the Atmel® AVR® ATmega128 as listed on page 76. In ATmega103 compatibility mode, Port C is output only, and the port C pins are not tri-stated when a reset condition becomes active.</p> <p>Note: The ATmega128 is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega128 as listed on page 77.</p>
Port E (PE7..PE0)	<p>Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port E also serves the functions of various special features of the ATmega128 as listed on page 80.</p>
Port F (PF7..PF0)	<p>Port F serves as the analog inputs to the A/D Converter.</p> <p>Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.</p> <p>The TDO pin is tri-stated unless TAP states that shift out data are entered.</p> <p>Port F also serves the functions of the JTAG interface.</p> <p>In ATmega103 compatibility mode, Port F is an input Port only.</p>
Port G (PG4..PG0)	<p>Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port G also serves the functions of various special features.</p> <p>The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>

In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32kHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in [Table 19 on page 50](#). Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

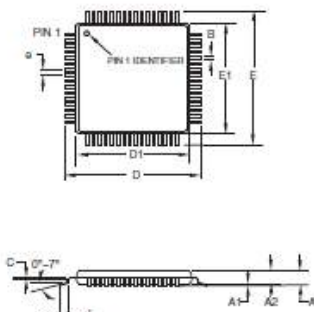
AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF

AREF is the analog reference pin for the A/D Converter.

PEN

PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. PEN has no function during normal operation.


COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	1.20	
A1	0.05	—	0.15	
A2	0.05	1.00	1.05	
D	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.30	—	0.45	
C	0.09	—	0.20	
L	0.45	—	0.75	
g	—	0.80 TYP	—	

Notes:

1. This package conforms to JEDEC reference MS-018, Variation AEB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

2010-10-00



2325 Orchard Parkway
San Jose, CA 95131

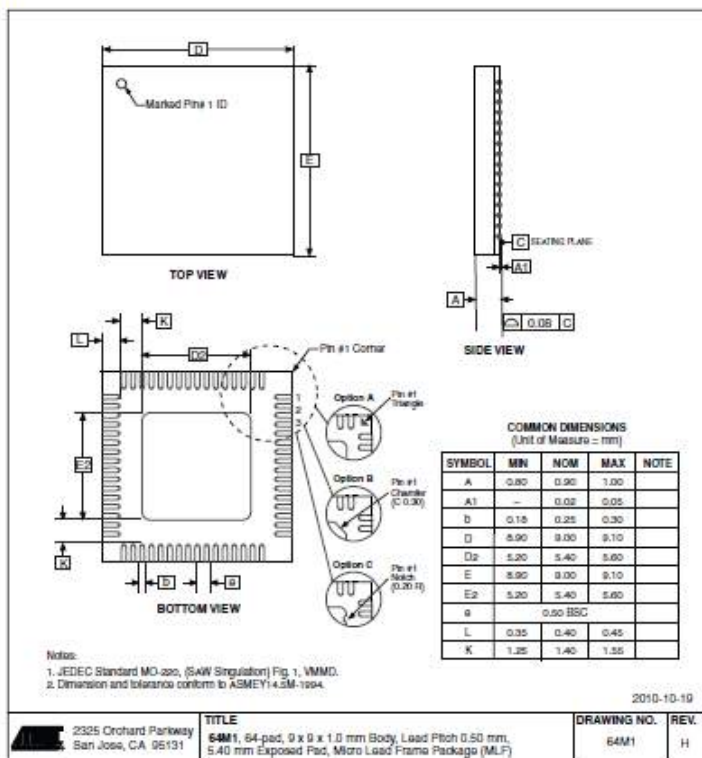
TITLE

64A, 64-load, 14 x 14 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Quad Flat Package (TQFP)

DRAWING NO.
REV.

64A

C

64M1


LAMPIRAN C

(DATA SHEET SENSOR FLOW TYPE G3/4")

29/9/2014

G3/4 Water Flow sensor - Wiki

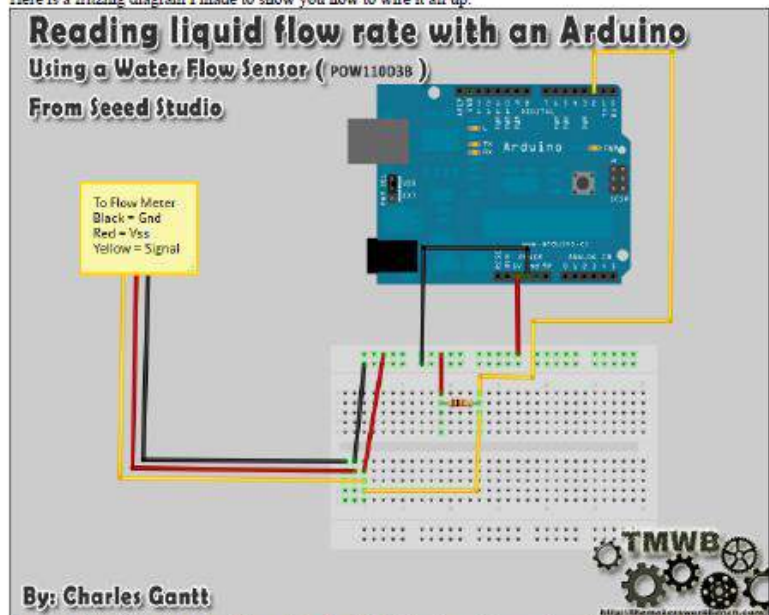


Specification

Mini. Working Voltage	DC 4.5V
Max. Working Current	15mA(DC 5V)
Working Voltage	5V~24V
Flow Rate Range	1~60L/min
Load Capacity	≤10mA(DC 5V)
Operating Temperature	≤80°C
Liquid Temperature	≤120°C
Operating Humidity	35%~90%RH
Water Pressure	≤2.0MPa
Storage Temperature	-25°C~+80°C
Storage Humidity	25%~95%RH

Mechanic Dimensions

Here is a fritzing diagram I made to show you how to wire it all up.



Once you have it wired up you will need to upload the following code to your Seeeduino. Once it is uploaded and you have some fluid flowing through the Water Flow Sensor, you can open the serial monitor and it will display the flow rate, refreshing every second.

Programming

```
// reading liquid flow rate using Seeeduino and Water Flow Sensor From Seeedstudio.com
// Code adapted by Charles Gantt from PC Fan RPM code written by Crenn @thebestcascenario.com
// http://thetmakersworkbench.com http://thebestcascenario.com http://seeedstudio.com

volatile int NtlopsFan; //measuring the rising edges of the signal
int Calc;
int hallsensor = 2; //The pin location of the sensor

void rpm () //This is the function that the interrupt calls
{
  NtlopsFan++; //This Function measures the rising and falling edge of the
  hall effect sensors signal
}

// The setup() method runs once, when the sketch starts
void setup() //
{
  pinMode(hallsensor, INPUT); //Initializes digital pin 2 as an input
}
```

http://www.seeedstudio.com/wiki/G34_Water_Flow_sensor

```

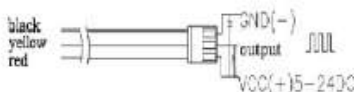
Serial.begin(9600); //This is the setup function where the serial port is
initialised,
attachInterrupt(0, rpm, RISING); //and the interrupt is attached
}
// The loop() method runs over and over again,
// as long as the Arduino has power
void loop ()
{
  NbTopsFan = 0; //Set NbTops to 0 ready for calculations
  sei(); //enables interrupts
  delay (1000); //Wait 1 second
  cli(); //Disable interrupts
  Calc = (NbTopsFan * 60 / 5.5); //(Pulse frequency x 60) / 5.5Q, = flow rate
  //in L/hour
  Serial.print (Calc, DEC); //Prints the number calculated above
  Serial.print (" L/hour\r\n"); //Prints "L/hour" and returns a new line
}

```

You can refer our forum for more details about Reading Water Flow rate with Water Flow Sensor (<http://www.seedstudio.com/forum/viewtopic.php?f=4&t=989&p=3632#p3632>) .

Wiring Diagram

The external diameter of thread the connections use is 1.4mm.



Output Table

Pulse frequency (Hz) in Horizontal Test= 5.5Q, Q is flow rate in L/min. (Results in +/- 3% range)

Output pulse high level	Signal voltage >4.5 V(input DC 5 V)
Output pulse low level	Signal voltage <0.5V(input DC 5V)
Precision	3% (Flow rate from 1L/min to 10L/min)
Output signal duty cycle	40%~60%

LAMPIRAN C
(*MANUAL BOOK POMPA*)

PENTING

Jarak Minimum
A=30 Cm
B=30 Cm
C=30 Cm

4. **Pilih** tempat dimana pompa harus dipasang dengan mudah dapat diakses atau diperbaiki setelah pemasangannya. Untuk pemasangan dipertajam yang sempat dibutuhkan namun seperti ditunjukkan pada Gb.2.
5. Kurangi jumlah tikungan-tikungan dalam sambungan pipa untuk mencegah terjadinya kebocoran dalam instalasi pipa, sehingga dapat mengurangi hambatan alir.

6. Ukur kedalaman permukaan air dalam sumur sampai ke permukaan tanah di atasnya. Standar tinggi hisap 8 m (maksimal 9 m).

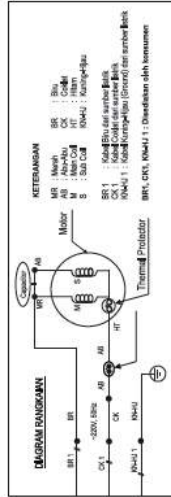
4. SPESIFIKASI POMPA

Model	PS-130 BIT			PS-135 E		
U: 1x220 V-50Hz	H: 20 - 10 meter	Pressure Switch On: 1,1 Kg/cm ² On: 1,3 Kg/cm ²		U: 1x220 V-50Hz	H: 20 - 5 meter	Pressure Switch On: 1,1 Kg/cm ² On: 1,3 Kg/cm ²
8,9F 370 V-1.2 A	M: Max: 9 meter	Pipa H=ap = 1"		8,9F 370 V-1.3 A	M: Max: 9 meter	Pipa H=ap = 1"
N: 2500 m ³ /h FX4	Temp air: Max: 40 °C	Pipa doring = 1"		N: 2500 m ³ /h FX4	Temp air: Max: 40 °C	Pipa doring = 1"

5. CARA MENGOPERASIKAN POMPA

1. Pompa ini membutuhkan sedikit air pascipan pada saat pertama kali dipaparkan.
2. Tuaran air pascipan sedikit lebih banyak pascipan sangat rumah pompa lebih penuh oleh air, tetapi akan berubah menjadi pascipan (Hopper Cap).
3. Pasang penutup lubang pascipan (Hopper Cap) dan tutup semua kran.
4. Hidupkan pompa air listrik, sebenarnya tunggulah beberapa saat sampai pompa menghisap air, kemudian buka kran sedikitnya, dan air akan mengalir.
5. Jika air tidak keluar, mungkin disebabkan oleh kurangnya air pascipan, jadi dingkatlah tahap pascipan tersebut.

6. CARA PENYAMBUNGAN KABEL



- [illegible]

2. BENTUK DAN NAMA-NAMA KOMPONEN POMPA

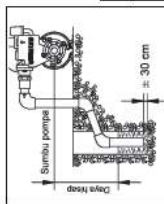
Diagram of a hand-operated sprayer with the following labeled parts:

- Pressure Switch
- Trigger Jockey (Small Tank)
- Safety/Jockey Separator
- Safety/Jockey Valve
- Motor
- Topup Inlet
- Remedy Pencil
- Fibre Head
- Penitup Latching Partridge (Upper Case)
- Fibre Lockdown Air

3. PETUNJUK PENASANGAN POMPA

1. Periksa kedalaman sumur dan pastikan daya hisap pompa sesuai dengan kemampuan pompa/juga memperimbangkan penurunan permukaan air pada musim kemarau.
2. Pasang pompa sedekat mungkin dengan sumur.

Daya hisap (m)	8	7,5	7	6,5	6
Panjang (horizontal) max. Pipa hisap	9	13,5	18	22,5	27



- Pompa harus dileakkan pada ketinggian ± 30 cm di atas permukaan lantai/tanah dan permukaan harus rata agar pompa tidak terendam air.

LAMPIRAN E

PERHITUNGAN KALIBRASI SENSOR

No	Pembacaan STD(X)	Pembacaan Alat		Rata- Rata	Koreksi (Y)	X^2	$Y-Y'$
		RataNaik	RataNaik				
1	3	3,69	3,73	3,71	-0,71	9	-0,272
2	5	5,53	5,71	5,62	-0,62	25	-0,182
3	7	7,38	7,47	7,425	-0,425	49	0,013
4	9	9,22	9,35	9,285	-0,285	81	0,153
5	11	11,07	11,23	11,15	-0,15	121	0,288
JMLH	35	36,89	37,49	37,19	-2,19	285	0
Rata2	7	7,378	7,498	7,438	-0,438	57	0

$X*Y$	$(Y-Y')^2$	Y_{reg}	$(Y-Y_{reg})^2$	$H(I)$	$H(I)/X$
-2,13	0,073984	-0,31241	0,158079	0,04	0,013333
-3,1	0,033124	-0,3752	0,059925	0,18	0,036
-2,975	0,000169	-0,438	0,000169	0,09	0,012857
-2,565	0,023409	-0,5008	0,046568	0,13	0,014444
-1,65	0,082944	-0,56359	0,171058	0,16	0,014545
-12,42	0,21363	SSR	0,435799	0,6	0,0911804
-2,484	0,042726			0,12	0,0182361

BIODATA PENULIS



Shaski Icraini Diotama merupakan nama lengkap penulis dengan nama panggilannya, Shaski. Penulis dilahirkan di Surabaya pada tanggal 23 Juni 1997. Penulis merupakan anak kedua dari tiga bersaudara. Pada saat ini penulis tinggal di Jalan Petemon Barat No. 248, Kecamatan Sawahan, Surabaya. Pada tahun 2009, penulis menyelesaikan pendidikan tingkat dasar di SDN Petemon XI Surabaya. Pada tahun 2012 penulis menyelesaikan pendidikan tingkat menengah pertama di SMP Negeri 43 Surabaya. Tahun 2015 berhasil menyelesaikan pendidikan tingkat menengah atas di SMA Negeri 17 Surabaya. Dan pada tahun 2018 ini, penulis mampu menyelesaikan gelar ahli madya di Program Studi DIII Teknik Instrumentasi, Departemen Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis berhasil menyelesaikan Tugas Akhir dengan judul **“PERANCANGAN SISTEM PENGENDALIAN LAJU ALIRAN PADA ALAT UJI KEBOCORAN PIPA SEBAGAI PENUNJANG PRAKTIKUM SISTEM INSTRUMENTASI INDUSTRI”**. Bagi pembaca yang memiliki kritik dan saran mengenai tugas akhir ini, dapat menghubungi penulis melalui email shaskiicraini23@gmail.com.